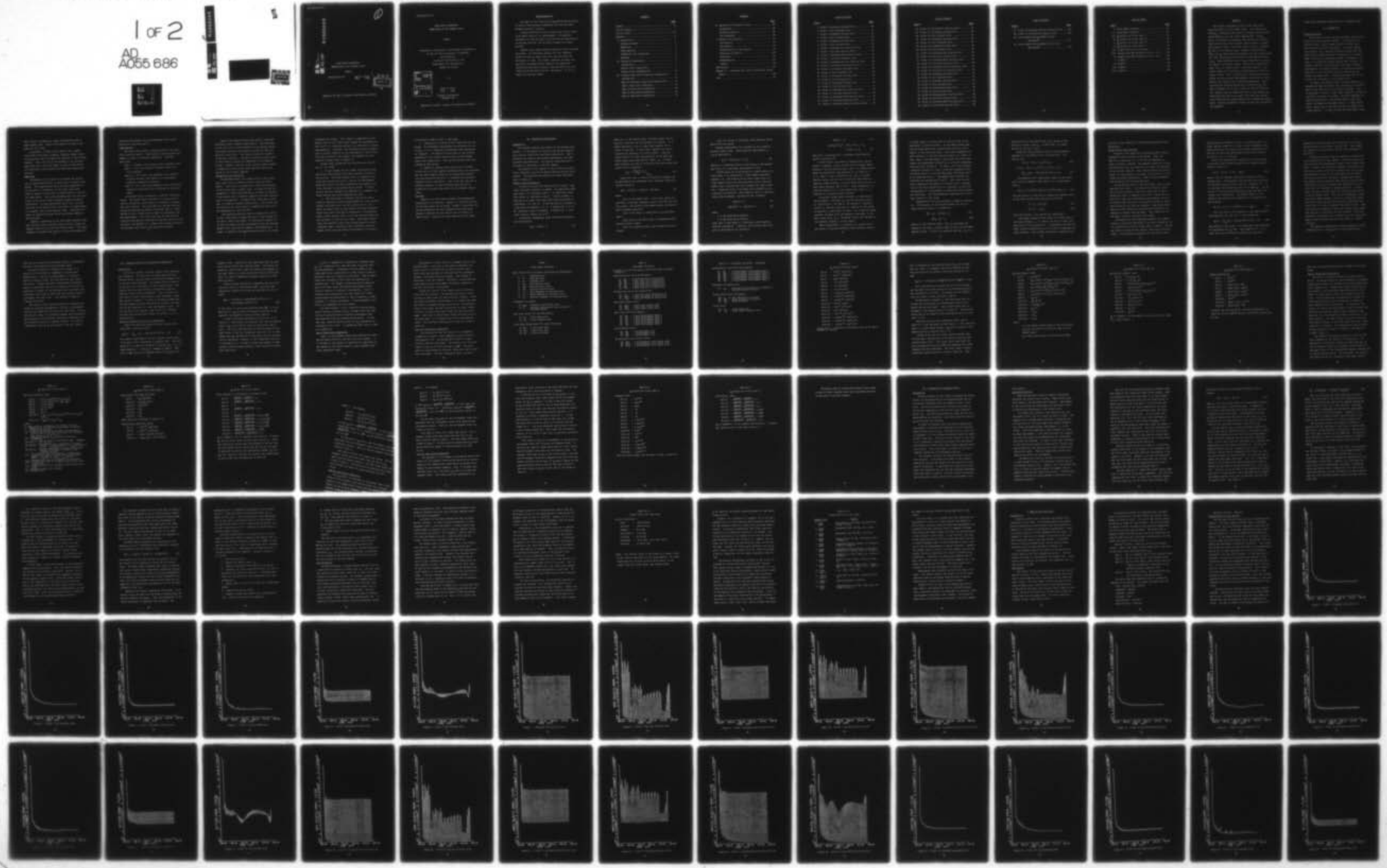
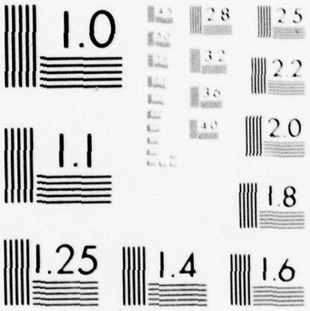
AD-A055 686 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 17/7
STATE NOISE COVARIANCE COMPUTATION IN THE KALMAN FILTER.(U)
DEC 77 D A ARPIN
UNCLASSIFIED AFIT/GE/EE/77-42 NL

1 OF 2
AD
A055 686





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

AFIT/GE/EE/77-42

①

AD A055686

AD No.

DDC FILE COPY

STATE NOISE COVARIANCE COMPUTATION IN THE KALMAN FILTER

THESIS

AFIT/GE/EE/77-42

David A. Arpin
Capt USAF

DDC
RECEIVED
JUN 22 1978
E

Approved for public release; distribution unlimited

78 06 13 182

AFIT/GE/EE/77-42

STATE NOISE COVARIANCE
COMPUTATION IN THE KALMAN FILTER

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

ACCESSION for
NTIS White Section ☒
DDC Buff Section ☐
UNANNOUNCED ☐
JUSTIFICATION ....
....
BY ....
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

by

David A. Arpin
Capt USAF

Graduate Engineering
December 1977

Approved for public release; distribution unlimited.

Acknowledgements

The idea for this thesis was suggested by Maj Ken Myers of the Air Force Avionics Laboratory, who also provided valuable technical insights.

I would particularly like to express my sincere thanks to my thesis advisor, Dr. Pete Maybeck. His generous investment of time and effort to criticize and encourage me, to give me direction, and to assist me made this thesis possible.

Several other people deserve mention for their valuable assistance. My laboratory sponsor, Mr. Bill Shephard, helped to focus my vague ideas into specific goals and directions of study. My readers, Captains Jim Negro and Gary Reid, provided helpful comments and suggestions. Significant additional technical help and advice was provided by Capt Ron Butler and Mr. Stan Musick. To all of these, my sincerest thanks.

Contents

Page

Preface ii

List of Figures v

List of Tables vii

Abstract ix

I. Introduction 1

Problem Statement 1

Objectives 2

Study Approach 3

Assumptions and Limitations 4

Overview 6

II. Theoretical Background 7

Introduction 7

Kalman Filter Formulation 7

Kalman Filter Implementation 13

III. Proposed State Noise Covariance Computations 17

Introduction 17

Need for Alternate State Noise Computations 17

Type I State Noise Computation 19

Type III State Noise Computation 20

Type IV State Noise Computation 29

Type II State Noise Computation 33

Contents

Page

IV. Evaluation of Proposed Filters 38

Introduction 38

Covariance Analysis 39

Test Parameters 46

V. Results and Discussion 53

Introduction 53

Test Results 53

Interpretation of Test Results 55

Discussion of Results 104

Conclusion 112

Recommendations 112

Summary 119

Bibliography 121

Appendix A: Integrated GPS Inertial Navigation System

Models 122

Vita 137

List of Figures

Figure Page

1 Filter I Estimated Longitude Error 56
2 Filter I True Longitude Error 57
3 Filter I Estimated Latitude Error 58
4 Filter I True Latitude Error 59
5 Filter I Estimated Altitude Error 60
6 Filter I True Altitude Error 61
7 Filter I Estimated East Velocity Error 62
8 Filter I True East Velocity Error 63
9 Filter I Estimated North Velocity Error 64
10 Filter I True North Velocity Error 65
11 Filter I Estimated Vertical Velocity Error 66
12 Filter I True Vertical Velocity Error 67
13 Filter II Estimated Longitude Error 68
14 Filter II True Longitude Error 69
15 Filter II Estimated Latitude Error 70
16 Filter II True Latitude Error 71
17 Filter II Estimated Altitude Error 72
18 Filter II True Altitude Error 73
19 Filter II Estimated East Velocity Error 74
20 Filter II True East Velocity Error 75
21 Filter II Estimated North Velocity Error 76
22 Filter II True North Velocity Error 77
23 Filter II Estimated Vertical Velocity Error ... 78

List of Figures

Figure Page

24 Filter II True Vertical Velocity Error 79

25 Filter III Estimated Latitude Error 80

26 Filter III True Latitude Error 81

27 Filter III Estimated Longitude Error 82

28 Filter III True Longitude Error 83

29 Filter III Estimated Altitude Error 84

30 Filter III True Altitude Error 85

31 Filter III Estimated East Velocity Error 86

32 Filter III True East Velocity Error 87

33 Filter III Estimated North Velocity Error 88

34 Filter III True North Velocity Error 89

35 Filter III Estimated Vertical Velocity Error .. 90

36 Filter III True Vertical Velocity Error 91

37 Filter IV Estimated Longitude Error 92

38 Filter IV True Longitude Error 93

39 Filter IV Estimated Latitude Error 94

40 Filter IV True Latitude Error 95

41 Filter IV Estimated Altitude Error 96

42 Filter IV True Altitude Error 97

43 Filter IV Estimated East Velocity Error 98

44 Filter IV True East Velocity Error 99

45 Filter IV Estimated North Velocity Error 100

46 Filter IV True North Velocity Error 101

List of Figures

Figure Page

47 Filter IV Estimated Vertical Velocity Error ... 102

48 Filter IV True Vertical Velocity Error 103

49 Error Growth with $\underline{Q}_d$ Added for Each Sub-Interval 117

50 Error Growth with $\underline{Q}_d$ Added Only for Last Sub-Interval 118

List of Tables

Table Page

I Filter Model Variables 21

II Truth Model Variables 22

III Q_d Values for Filter Type I 24

IV Q_d Values for Filter Type III 26

V Q_d Values for Filter Type IV 30

VI Q_d Values for Filter Type II 35

VII Flight Profile for Test Cases 49

VIII Steady State RMS Estimation Errors, $\Delta t=2$ 105

IX Steady State RMS Estimation Errors, $\Delta t=0.2$ 106

X F Matrix 124

XI Q Matrix 132

XII H Matrix 133

XIII R Matrix 136

Abstract

This report investigates forms of the state noise covariance matrix in the Kalman Filter. This matrix, denoted $\underline{Q}_d$, incorporates the effects of random errors driving system dynamics into the filter computations. The $\underline{Q}_d$ matrix is derived by integration from the matrix of continuous time driving noise strengths, which normally includes only diagonal terms. This often leads to use of a diagonal $\underline{Q}_d$ matrix with constant terms. However, the derivation shows that $\underline{Q}_d$ should have off-diagonal and time varying terms. The study investigates the effects of including such terms in $\underline{Q}_d$. Three alternate forms of $\underline{Q}_d$ are derived for a specific inertial navigation system. These, and a standard diagonal form, are tested using a covariance analysis. The results show little difference in performance for the different filters. This is attributed to two primary factors: highly accurate external measurements, and the use of integration sub-intervals for covariance propagation. These sub-intervals generate appropriate off-diagonal $\underline{Q}_d$ terms when a diagonal form of $\underline{Q}_d$ is used over each sub-interval. This suggests that an appropriate form of non-diagonal $\underline{Q}_d$, which would not have to be added in at each sub-interval, could significantly reduce Kalman Filter computation requirements. Specific additional studies to test this possibility are suggested.

STATE NOISE COVARIANCE COMPUTATION IN THE KALMAN FILTER

I. Introduction

Problem Statement

The Air Force mission requires aerospace vehicles with highly accurate navigation systems. To achieve the required accuracy, these systems typically include two or more separate navigation subsystems. The usual combination consists of an inertial navigation system (INS) and some external measuring device or devices such as Doppler Radar, Tactical Air Navigation (TACAN), a barometric altimeter, or a Global Positioning System (GPS) receiver. By properly combining the information from these different measurements, the onboard computer can generate navigation data which is more accurate than that supplied by any single instrument alone. A widely used algorithm for combining this information is the Kalman Filter.

In theory, the Kalman Filter is exactly defined by a mathematical model which describes the behavior and performance of the vehicle and its navigation systems. This "truth model" includes a large number (typically 50-100) of individual variables or "states", including components of position and velocity, INS platform angles, gyro drift rates, and measuring instrument biases. In most applications, states such as position and velocity exhibit nonlinear behavior. In order to keep the truth model linear, these states are modeled as linear perturbations about a known nominal path. These "error states" are used in the model for this study.

In practice, a Kalman Filter based on this complex truth model would require computer resources (memory space and processing time) far beyond the capacity of any airborne computer. Thus, the task of the designer is to introduce approximations to the truth model in order to meet practical constraints, while maintaining sufficient navigational accuracy.

Objectives

The purpose of this study is to examine one specific type of approximation which is widely used in Kalman Filter design. This approximation deals with the computation of the strength, or covariance, of the state noise (also known as dynamic driving noise or system noise) in the filter computations. This noise determines how fast the uncertainty in the vehicle state increases between navigation measurements. In many applications which use a discrete time form of the Kalman Filter, this matrix is approximated as a diagonal matrix with constant terms. Individual elements are adjusted for good filter performance through a tuning process.

However, examination of the truth model equations show that the state noise covariance matrix should include off-diagonal terms and vary with the vehicle state. This study attempts to determine the effects on filter performance of restoring off-diagonal and state dependent terms to the state noise covariance matrix.

Study Approach

The study tested several alternate forms of the state noise covariance matrix. These matrices were assigned type numbers in order of increasing complexity. The forms tested were:

Type I - Diagonal, with constant terms (standard type).

Type II - Full (including off-diagonal terms), but still constant.

Type III - Full, with state dependent terms derived directly from the truth model by an approximate numerical integration technique.

Type IV - Full, with state dependent terms analytically derived to approximate the performance of the truth model.

To evaluate these alternatives, a specific system was tested. The system chosen consisted of a fairly typical inertial navigation system aided by a barometric altimeter and a GPS receiver. The truth model for the baro-inertial system was developed by Widnall and Grundy (Ref 1). Myers and Butler (Ref 2) modified this model to incorporate the GPS measurements. The filter model was derived directly from the truth model by deleting all but the first 16 of the 52 states (see Appendix A). Kalman Filters based on each of the proposed noise matrix types were then designed.

Each of the proposed filters was tuned to give good performance over a chosen flight profile. The covariance of the true error in the estimation of the primary navigation states (position and velocity errors) was then computed for each filter. This type of testing is termed a "covariance analysis". The error covariance results serve as the primary criterion for comparison of the proposed filters. Another criterion is the relative burden (number of computations and storage space required) which each places on the airborne computer.

Assumptions and Limitations

As mentioned above, the system tested in this study uses error states to maintain linear truth and filter models. These errors are modeled as linear perturbations to the states about a nominal path. In order to keep the assumption of linear perturbations as accurate as possible, a new nominal path is normally computed each time a measurement is incorporated. A Kalman Filter using these techniques is termed an Extended Kalman Filter. This filter uses the values of the errors which it computes for each external measurement to reset the INS.

The covariance analysis used to evaluate the filters in this study works in theory only for a linear Kalman Filter and a linear truth model. As described in Chapter IV, it does not supply external measurements which the Extended Kalman Filter could use to compute a new nominal path. Instead, it supplies a precomputed nominal path to the filter throughout the flight. Thus, errors in computation of the nominal path by the filter cannot degrade the performance of the filter in a covariance analysis, as they would in actual implementation. Therefore, the covariance analysis presents only a limited indication of the performance of each filter. This type of analysis was judged to be adequate for this study for reasons given in Chapter IV.

Several other assumptions and limitation are significant in this study:

1. It was assumed that the Global Positioning System satellites always maintain the same positions relative to the aircraft. The extra program logic to propagate satellites and periodically select a new set for measurements would have greatly increased computation requirements and is not relevant to the problem being studied. A single case of satellite geometry was computed for a randomly selected time and then used throughout the study.

2. The truth model as obtained from the Air Force Avionics Laboratory contains known anomalies in the vertical channel (altitude error, vertical velocity error, etc.) which can cause errors in that channel to become excessive. These errors were corrected by Intermetrics, Incorporated under an AFAL contract (Ref 3), but the corrected model was not available in time for this study. Some simple corrections were applied in the Filter tuning process to obtain a workable model. Because of this limitation, vertical channel states were not tuned and evaluated as critically

as horizontal channel states in the study.

3. Exhaustive fine tuning of the filters was not conducted. No performance requirements are available for the filters, so it is impossible to determine what performance is "adequate". The important factor here is the relative peformance of the filters based on each of the proposed state noise computations. This can be evaluated with only reasonably good tuning.

4. Numerical problems which often occur in Kalman Filters due to finite computer wordlength were not considered. The CDC Cyber computer used for the simulations provides very high precision (60 bits), so roundoff and truncation problems are not significant. This is not the case for an airborne computer, so these problems would have to be studied for an actual application of the proposed computations.

Overview

Chapter II of this report presents the mathematical background for the problem. Chapter III shows derivations for the four models to be tested. Chapter IV describes the testing method and the test parameters which were used. Chapter V presents the test results and an analysis of these results. Because the truth model and filter model

II. Theoretical Background

Introduction

This chapter presents the context of the problem being studied. It introduces the notation used and the theoretically correct state noise covariance computation. Then it discusses the need for approximate computations, and some available approximations. The specific approximations to be tested are described in the next chapter.

The purpose of this chapter is not to derive the equations rigorously, but only to depict them and explain their context. The equations and notation are taken from Reference 4, Chapter 4.

Kalman Filter Formulation

In order to discuss the Kalman Filter in detail, some vector and matrix notation is needed. An underscored upper case letter, $\underline{A}$, indicates a matrix. An underscored lower case letter, $\underline{x}$, indicates a vector. A matrix element is depicted as an upper case letter with indices enclosed in parentheses, e.g. P(5, 5). A time derivative is denoted by a dot above the quantity, e.g. $\dot{\underline{x}}$. A superscript T, as in $\underline{A}^T$, indicates a matrix transpose. A superscript -1 indicates a matrix inverse, as in $\underline{A}^{-1}$.

A nonlinear, homogeneous state differential equation can be written as

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t), t] \qquad (1)$$

where $\underline{x}(t)$ is the system state, including states such as position, velocity, and INS platform tilt angles. It is highly desirable to use a linear state equation for a filter if possible. In many cases, equation (1) can be linearized by changing $\underline{x}$ to "error states". Then the states consist of linear perturbations to the above mentioned states about some nominal "path", $\underline{x}_n$. A linear state equation can then be written for the errors about the nominal path by computing

$$\underline{F}(t) = \left. \frac{\partial \underline{f}[\underline{x}(t), t]}{\partial \underline{x}} \right|_{\underline{x}=\underline{x}_n} \qquad (2)$$

Using this form, a class of stochastic processes can be described by the continuous time, stochastic state differential equation

$$\dot{\underline{x}}(t) = \underline{F}(t)\underline{x}(t) + \underline{B}(t)\underline{u}(t) + \underline{G}(t)\underline{w}(t) \qquad (3)$$

where:

- $\underline{x}(t)$ is the system state. Since error states are used here, $\underline{x}$ includes components such as position and velocity errors, INS platform misalignment angles, and measurement instrument anomalies.

- $\underline{F}(t)$ is the plant or system matrix, as described above.

- $\underline{B}(t)$ and $\underline{u}(t)$ are terms used to incorporate deterministic control inputs.

- $\underline{G}(t)$ is a selection matrix for stochastic driving noises.

- $\underline{w}(t)$ is a vector of zero-mean, white Gaussian noises which drive the system.

External measurements are available to the system at discrete sample times. These imperfect measurements, $\underline{z}$, can be described as

$$\underline{z}(t_i) = \underline{H}(t_i)\underline{x}(t_i) + \underline{v}(t_i) \tag{4}$$

where $\underline{H}$ is a selection matrix of the states in the measurement and $\underline{v}$ is a zero-mean, white Gaussian noise.

A white noise can be described as a random process for which there is no correlation in time between subsequent samples; i.e., the process can go from a known value at a given sample time to any other possible value at the next sample time, no matter how close together the sample times are. A Gaussian noise is one whose joint probability distributions are Gaussian. White Gaussian noise is characterized by two parameters, the mean and the "strength":

$$E(\underline{w}(t)) = \underline{0} \tag{5}$$

$$E(\underline{w}(t)\underline{w}^T(t')) = \underline{Q}(t)\delta(t-t') \tag{6}$$

where

E is the expectation operator,

δ is the Dirac delta function.

Thus, $\underline{w}(t)$ is described as a "zero-mean, white Gaussian noise of strength $\underline{Q}$". Similarly, the discrete time noise $\underline{v}(t)$ is described by its statistics:

$$E(\underline{v}(t)) = \underline{0} \tag{7}$$

$$\begin{aligned} E(\underline{v}(t_i)\underline{v}(^T(t_j)) &= R(t_i) \text{ for } t_i = t_j \\ &= \underline{0} \text{ for } t_i \neq t_j \end{aligned} \tag{8}$$

Then $\underline{v}(t_i)$ is described as a "zero-mean, white Gaussian noise of covariance $\underline{R}$".

In this thesis, the effect of deterministic control inputs is ignored, so $\underline{B}(t)$ and $\underline{u}(t)$ will no longer appear. Because of the driving noise $\underline{w}(t)$, $\underline{x}(t)$ is a random process. Under the assumptions that equation (3) is linear in $\underline{x}$ (i.e., that $\underline{F}$ is not a function of $\underline{x}$), $\underline{w}(t)$ is a white Gaussian noise, and $\underline{x}(t)$ can initially be described as a Gaussian random variable, it can be shown that $\underline{x}(t)$ will always remain Gaussian (Ref 4:4-22). These assumptions all hold for equation (3), so the system error state, $\underline{x}$, is modelled as a Gaussian random variable.

A Gaussian random vector variable $\underline{x}$ is described by two parameters - the mean, $\underline{m}_x$, and the covariance matrix $\underline{P}$. Heuristically, the mean is the average or "expected" value of $\underline{x}$, and the covariance is a measure of the spread of possible $\underline{x}$ values about the mean. For a Gaussian random variable, the mean value corresponds to the mode, or the most likely value of $\underline{x}$. The Kalman filter uses the mean as its estimate, denoted $\hat{\underline{x}}$, of the state $\underline{x}$.

When a measurement is processed in a Kalman Filter, the result is the best obtainable state estimate relative to a wide range of criteria (Ref 4: 5-49 to 5-54), so the covariance is relatively small. As the vehicle moves away from the measurement time, uncertainties build up in the states, so the covariance grows. The filter propagates the state estimate, $\underline{\hat{x}}$, and the covariance matrix, $\underline{P}$, forward in time. When a new measurement is processed, the filter must combine the measurement information, $\underline{z}$, with the propagated state estimate, denoted $\underline{\hat{x}}$-. In effect, the filter computes a weighted average of $\underline{\hat{x}}$- and $\underline{z}$, using the propagated state covariance matrix $\underline{P}$-, and the measurement covariance matrix $\underline{R}$, as weighting factors. The updated state estimate and covariance are denoted as $\underline{\hat{x}}^+$ and $\underline{P}^+$. Thus, there are two basic computations performed in a Kalman Filter: propagation of the state estimate and its covariance in time, and incorporation of a measurement to update the state estimate and covariance. The time propagation equations are of primary interest in this study.

Propagation of the state estimate is based on equation (3). This solution is facilitated through the use of a state transition matrix, $\underline{\Phi}$, defined by:

$$\underline{\dot{\Phi}}(t,\ t_0) = \underline{F}(t)\underline{\Phi}(t,\ t_0) \qquad (9)$$

$$\underline{\Phi}(t_0,\ t_0) = \underline{I} \qquad (10)$$

where $\underline{I}$ is the identity matrix. $\underline{\Phi}(t,\ t_0)$ describes the change in the state $\underline{x}$, between times t_0 and t for the homogeneous system. If $\underline{F}(t)$ is a constant, then $\underline{\Phi}(t,\ t_0)$ is a function only of the elapsed time $t-t_0$, and not of the particular values t and t_0. In this case, $\underline{\Phi}$ is often written as $\underline{\Phi}(t-t_0)$ or $\underline{\Phi}(\Delta t)$.

Using this state transition matrix, the propagation equations for the Kalman Filter can be derived. The results are:

$$\underline{\hat{x}}^{-}(t_i) = \underline{\Phi}(t_i, t_{i-1})\underline{\hat{x}}^{+}(t_{i-1}) \tag{11}$$

$$\begin{aligned}\underline{P}^{-}(t_i) &= \underline{\Phi}(t_i, t_{i-1})\ \underline{P}^{+}(t_{i-1})\ \underline{\Phi}^{T}(t_i, t_{i-1}) \\ &\quad + \int_{t_{i-1}}^{t_i} \underline{\Phi}(t_i, \tau)\ \underline{G}(\tau)\underline{Q}(\tau)\underline{G}^{T}(\tau)\underline{\Phi}^{T}(t_i,\tau)d\tau\end{aligned} \tag{12}$$

The equations for a measurement update use the values of $\underline{P}^{-}$ and $\underline{R}$ to compute a gain matrix, $\underline{K}$, for measurement time t_i:

$$\underline{K}(t_i) = \underline{P}^{-}(t_i)\underline{H}^{T}(t_i)[\underline{H}(t_i)\underline{P}^{-}(t_i)\underline{H}^{T}(t_i)+\underline{R}(t_i)]^{-1} \tag{13}$$

This gain matrix is then used to compute the updated values of the state estimate and covariance as follows:

$$\underline{\hat{x}}^{+} = \underline{\hat{x}}^{-} + \underline{K}(\underline{z}-\underline{H}\underline{\hat{x}}^{-}) \tag{14}$$

$$\underline{P}^{+} = \underline{P}^{-} - \underline{K}\ \underline{H}\ \underline{P}^{-} \tag{15}$$

(The time indices t_i are omitted for convenience.)

The primary purpose of the filter is to keep track of the state estimate, $\underline{\hat{x}}$. In order to do this, it must also keep track of the covariance, $\underline{P}$. As equations (13) and (14) show, accurate evaluation of $\underline{P}$ is critical in computing the state estimate. Thus, proper evaluation of equations (12) and (15) is very important in achieving good Kalman Filter performance.

Kalman Filter Implementation

A typical truth model for an aerospace aided interial navigation system contains 50-100 states. Thus, the matrices of equations (3) to (15) are of dimensions as large as 50x50 to 100x100. The computer resources to store and manipulate such large matrices are simply not available in the airborne computer. Many simplifications and approximations to these models can be made, often with only a very slight performance degradation. Some of these implementation techniques will be discussed here.

Of the 50-100 states in the truth model, some are more important than others. The position and velocity states are of direct interest to the user, since that is what a navigation sytem is supposed to tell him. States such as gyro drift rates and accelerometer misalignments are only of interest when they help the navigation system to obtain better position and velocity estimates. Many states which have only small effects on navigation performance can be combined or dropped from the model. This simplified "filter model" typically contains 10-20 states.

The theoretical and laboratory analysis of a system yields an accurate system model in the form of equation (3), i.e. involving an $\underline{F}$ matrix rather than a Φ matrix. The Kalman Filter is based on the $\underline{\Phi}$ matrix, which can be derived from $\underline{F}$. The solution to equations (9) and (10) is very time consuming to generate on line. If the $\underline{F}(t)$ matrix is slowly varying, then for a Δt that is small compared to the time constants of the system, the solution to equations (9) and (10) can often be approximated to sufficient accuracy by $\underline{\Phi}(t_i+\Delta t,\ t_i) = \underline{I} + \underline{F}(t_i)\Delta t$.

Using this approximation, equation (3) can be written in discrete time difference equation form (ignoring deterministic inputs) as:

$$\underline{x}(i+1) = \underline{\Phi}(i+1,\ i)\ \underline{x}(i) + \underline{W}_d(i) \qquad (16)$$

where i and i+1 indicate consecutive instants of time Δt seconds apart. The $\underline{w}_d$ term is the discrete time equivalent of the white Gaussian driving noise $\underline{w}$ in equation (3). It is zero-mean, white Gaussian noise of strength $\underline{Q}_d$. The subscript "d" serves to distinguish $\underline{Q}_d$ and $\underline{w}_d$ from their continuous time counterparts. The primary impact of this change on the Kalman Filter equations is in equation (12), which becomes

$$\underline{P}(i+1) = \underline{\Phi}(i+1,\ i)\ \underline{P}(i)\underline{\Phi}^T(i+1,\ i) + \underline{Q}_d(i) \qquad (17)$$

The first term in this equation is the same as before. From equations (12) and (17), it can be seen that

$$\underline{Q}_d(i) = \int_{t_{i-1}}^{t_i} \underline{\Phi}(t_i,\ \tau)\underline{G}(\tau)\underline{Q}(\tau)\underline{G}^T(\tau)\underline{\Phi}^T(t_i,\ \tau)d\tau \qquad (18)$$

The purpose of this study is to investigate some techniques for evaluating this $\underline{Q}_d$ term. From equation (17), it is clear that this term directly affects the covariance computation, which was seen to be important for accurate state estimation in equations (13) and (15). Thus, numerical computation of $\underline{Q}_d$ directly affects the performance of the filter.

A direct evaluation of $\underline{Q}_d$ by solution of equation (18) is not appropriate for two reasons. First, this would present an excessive computational burden to the airborne computer. Second, and a more basic problem, is that this will not give a correct result for a reduced order filter model. Many important states in the truth model are affected by the driving noise only indirectly, through other states. For example, the driving noise may cause the position error covariance to grow by increasing the uncertainty in velocity, which is integrated to compute position. In other cases, the best available model for a driving noise may be a time correlated noise. This can be incorporated in equation (3) by adding another state to $\underline{x}$, driving that state with white noise, and then driving other states with that state (this is called a shaping filter (Ref 4:4-80)). When such states are removed to obtain the reduced filter model, some sources of uncertainty are also removed. The direct evaluation of equation (18) to obtain $\underline{Q}_d$ in this case would model a time correlated noise as zero, which is clearly incorrect.

The designer compensates for this by adding so called "pseudo-noises" to the $\underline{Q}_d$ matrix in the Kalman Filter.

These are white noises with strengths chosen to approximate the error contributions of the discarded states.

The usual technique in Kalman Filter design is to treat all of the noises as independent, discrete-time pseudo-noises. Thus, $\underline{Q}_d$ is approximated as a diagonal matrix of constant terms, with one entry for each individual state. Through some type of performance evaluation, such as a covariance analysis or a Monte Carlo analysis as described in Chapter IV, the designer tests and adjusts the individual noise terms to obtain the best possible overall performance from the filter. This process is known as "tuning" the filter.

This type of filter design minimzes the computational burden of covariance propagation, since $\underline{Q}_d$ is precomputed and only one number must be stored for each state. The purpose of this study is to determine whether some alternative $\underline{Q}_d$ evaluations can give better navigation performance with an acceptable increase in computer loading. The specific alternative forms for $\underline{Q}_d$ are derived in the next chapter.

III. Proposed State Noise Covariance Computations

Introduction

In the last chapter, the basic Kalman Filter equations were presented. The need for an accurate state noise covariance matrix, $\underline{Q}_d$, was shown, and the usual diagonal form of the matrix was described. The purpose of this study is to evaluate some alternatives to this standard form. This chapter shows why some different computations might be expected to give better performance. Then it derives the four types of state noise matrices to be evaluated. These were listed in Chapter I in order of increasing complexity. The type numbers assigned there will be retained, but they will be described here in a more developmental order. The full $\underline{Q}_d$ matrix with constant terms (Type II) is derived from the time varying types (Types III and IV), so it is discussed last.

Need for Alternate State Noise Computations

Recall the defining equation for the discrete time state noise covariance matrix:

$$\underline{Q}_d(t_i) = \int_{t_{i-1}}^{t_i} \underline{\Phi}(t_i, \tau)\underline{G}(\tau)\underline{Q}(\tau)\underline{G}^T(\tau)\underline{\Phi}^T(t_i, \tau)d\tau \qquad (18)$$

The simplest approximation to this integral results from a first-order, Euler Integration of equation (18). Then $\underline{G}(t)$ $\underline{Q}(t)\underline{G}^T(t)$ is treated as constant in time and $\underline{\Phi}(t_i, t_{i-1})$ is approximated as $\underline{I}$. Then $\underline{Q}_d$ is computed as $\underline{GQG}^T\Delta t$. Thus, only if $\underline{GQG}^T$ has any off-diagonal terms will $\underline{Q}_d$ have off-diagonal terms. Physically, this would mean that the same noise source affects more than one state. Such noises are generally removed from a model by choosing the geometry of the model properly to avoid such direct noise correlations. Thus, this type of computation normally results in a diagonal $\underline{Q}_d$ matrix.

A more accurate evaluation of equation (18) can be obtained by letting $\underline{\Phi}$, and possibly $\underline{GQG}^T$, vary with time. Trapezoidal integration can then be used (Ref 4:6-113) to obtain

$$\begin{aligned}\underline{Q}_d(t) = 1/2[&\underline{\Phi}(t+\Delta t,\ t)\underline{G}(t)\underline{Q}(t)\underline{G}^T(t)\underline{\Phi}^T(t+\Delta t,\ t) \\ &+ \underline{G}(t+\Delta t)\underline{Q}(t+\Delta t)\underline{G}^T(t+\Delta t)]\Delta t \end{aligned} \tag{19}$$

In this form, it can be seen that even when $\underline{GQG}^T$ is a diagonal matrix, $\underline{Q}_d$ will have off-diagonal terms due to the generally non-symmetrical nature of the $\underline{\Phi}$ matrix.

If the first-order approximation for $\underline{\Phi}(t+\Delta t,\ t)$ of $\underline{I}+\underline{F}(t)\Delta t$ is substituted into equation (19), the off-diagonal terms in $\underline{Q}_d$ can be seen to come from the $\underline{F}(t)\Delta t$ term in $\underline{\Phi}(\Delta t)$. The usual justification for omitting these off-diagonal terms is that the integration step size (Δt) is small enough that the contribution of the $\underline{F}(t)\Delta t$ term in equation (19) is negligible. However, if the integration step size is increased (e.g., in order to reduce the number of computations required between updates), these terms will become more significant.

If $\underline{F}(t)$ is computed as a function of a nominal state, $\underline{x}_n$, (as in equation (2)), then the terms in $\underline{Q}_d$ will also be state-dependent. Furthermore, certain states in the truth model cause uncertainties in other states to grow at different rates as a function of the state. Some of these states are discarded in the filter and replaced with pseudo-noises. For example, g-sensitive and g^2-sensitive gyro drift rates cause the platform misalignment uncertainty to grow as a function of acceleration. Use of a constant value pseudo-noise for such states will either overestimtate the uncertainty in low g situations or underestimate it in high g situations. A state dependent noise matrix can compensate for this problem.

For these reasons, a full, state and time-varying $\underline{Q}_d$ matrix might be expected to give a Kalman Filter the capability of achieving better navigation performance than that obtained using the standard approximation. To test this idea, three alternative forms for $\underline{Q}_d$ computation are evaluated in this study. A standard $\underline{Q}_d$ type filter is used as a comparison.

Type I State Noise Computation

Type I is the standard state noise computation technique described briefly at the end of the last chapter. It is included in this study as a baseline for comparison for the results of the filters based on other proposed state noise computation types.

The $\underline{Q}_d$ matrix in this filter is a diagonal matrix with constant terms. Each term in this matrix represents the direct effect of driving noise on one particular state. These terms were adjusted to give good overall performance through a covariance analysis program. This program is explained in the next chapter. Basically, it uses both the truth model and the filter model to provide a measure of how well the filter is performing.

Tuning this type of filter tends to be an intuitive, trial and error process. The designer may have to trade accuracy in some states for more accuracy in others. Since the states are highly interrelated, it is seldom clear what the effect of changing one $\underline{Q}_d$ term will be. In the models used in this study, the most significant 16 states (Table I) from a 52 state truth model (Table II) form the filter model. Only four of these states contain driving noise in the truth model, so 12 pseudo-noises had to be added and tuned. The resulting tuned $\underline{Q}_d$ matrix terms are listed in Table III.

Type III State Noise Computation

This alternative uses the results of a trapezoidal integration to obtain a more complete $\underline{Q}_d$ matrix directly from equation (19). The $\underline{G}(t)\underline{Q}(t)\underline{G}^T(t)$ matrix is taken directly from the truth model. This matrix is in fact constant in time, so it can be written as $\underline{G}\underline{Q}\underline{G}^T$. The $\Phi(\Delta t)$ matrix is approximated as $\underline{I}+\underline{F}(t)\Delta t$, using the $\underline{F}$ matrix from the truth model. For each integration step, a value of

Table I

Filter Model Variables

Basic Pinson INS Error Model (East-North-Up Coordinates)

1. $\delta\lambda$ - Longitude Error
2. δL - Latitude Error
3. δh - Altitude Error
4. δV_E - East Velocity Error
5. δV_N - North Velcoity Error
6. δV_Z - Vertical Velocity Error
7. ε_E - East Component of Attitude Error
8. ε_N - North Component of Attitude Error
9. ε_Z - Vertical Component of Attitude Error

Altimeter Error Model

10. e_{hsf} - Altimeter Scale Factor Error
11. $\hat{\delta}_a$ - Vertical Acceleration Error Variable in Altitude Channel

User Clock States for the GPS Receiver

12. δr_p - Clock Phase Error
13. δr_b - Clock Frequency Bias

First Order Markov Model for Gyro Drift Rates

14. DX_f - X Gyro Drift Rate
15. DY_f - Y Gyro Drift Rate
16. DZ_f - Z Gyro Drift Rate

Table II

Truth Model Variables

Variables 1-16 are the same as the filter model variables (Table I).

G-Sensitive Gyro Drift Coefficients

17. DX_x - X gyro spin axis g-sensitivity
18. DX_y - X gyro input axis g-sensitivity
19. DY_x - Y gyro spin axis g-sensitivity
20. DY_y - Y gyro input axis g-sensitivity
21. DZ_y - Z gyro spin axis g-sensitivitity
22. DZ_z - Z gyro input axis g-sensitivity

G^2-Sensitive Gyro Drift Coefficients

23. DX_{xy} - X gyro spin input g^2-sensitivity
24. DY_{xy} - Y gyro spin input g^2-sensitivity
25. DZ_{yz} - Z gyro spin input g^2-sensivitity

Gyro Scale Factor Errors

26. GSF_x - X gyro scale factor error
27. GSF_y - Y gyro scale factor error
28. GSF_z - Z gyro scale factor error

Gyro Input Axis Misalignments

29. XG_y - X gyro misalignment about Y
30. XG_z - X gyro misalignment about Z
31. YG_x - Y gyro misalignment about X
32. YG_z - Y gyro misalignment about Z
33. ZG_x - Z gyro misalignment about X
34. ZG_y - Z gyro misalignment about Y

Accelerometer Biases

35. AB_x - X accelerometer bias
36. AB_y - Y accelerometer bias
37. AB_z - Z accelerometer bias

Accelerometer Scale Factor Errors

38. ASF_x - X accelerometer scale factor error
39. ASF_y - Y accelerometer scale factor error
40. ASF_z - Z accelerometer scale factor error

Table II - Truth Model Variables - Continued

Accelerometer Input Axis Misalignments

41. XA_y - X accelerometer misalignment about Y
42. XA_z - X accelerometer misalignment about Z
43. YA_x - Y accelerometer misalignment about X
44. YA_z - Y accelerometer misalignment about Z
45. ZA_x - Z accelerometer misalignment about X
46. ZA_y - Z accelerometer misalignment about Y

Barometric Altimeter Error

47. e_{po} - Error due to variation in altitude of a constant pressure surface.

Gravity Deflections and Anomaly

48. δ_{ge} - East deflection of gravity
49. δ_{gn} - North deflection of gravity
50. δ_{gz} - Gravity anomaly

Clock Errors

51. δr_a - Clock aging bias
52. δr_u - Clock random frequency bias

Table III

$\underline{Q}_d$ Values for Filter Type I

$Q(1,1) = 2\text{x}10^{-15}$ (rad^2/sec)

$Q(2,2) = 2\text{x}10^{-15}$ (rad^2/sec)

$Q(3,3) = 1000$ (ft^2/sec)

$Q(4,4) = .01$ (ft^2/sec^3)

$Q(5,5) = .01$ (ft^2/sec^3)

$Q(6,6) = .01$ (ft^2/sec^3)

$Q(7,7) = 2.5\text{x}10\text{-}10$ (rad^2/sec)

$Q(8,8) = 2.5\text{x}10^{-10}$ (rad^2/sec)

$Q(9,9) = 2.5\text{x}10^{-10}$ (rad^2/sec)

$Q(10,10) = 4\text{x}10^{-8}$ (1/sec)

$Q(11,11) = 5\text{x}10^{-4}$ (ft^2/sec^5)

$Q(12,12) = 400$ (ft^2/sec)

$Q(13,13) = 1\text{x}10^{-10}$ (ft^2/sec^3)

$Q(14,14) = 5.86\text{x}10^{-20}$ (rad^2/sec^3)

$Q(15,15) = 5.86\text{x}10^{-20}$ (rad^2/sec^3)

$Q(16,16) = 1.62\text{x}10^{-17}$ (rad^2/sec^3)

The $\underline{Q}_d$ matrix is obtained by multiplying each of the above terms by Δt seconds.

$\underline{F}(t)$ is computed for the starting value of $\underline{x}_n$, as in equation (2), then $\underline{F}$ is assumed to be only a function of time. The multiplication of equation (19) to be carried out is then

$$\underline{Q}_d(t_i) = 1/2[(\underline{I}+\underline{F}(t_i)\Delta t)\underline{G}\underline{Q}\underline{G}^T(\underline{I}+\underline{F}(t_i)\Delta t)^T+\underline{G}\underline{Q}\underline{G}^T]\Delta t \quad (20)$$

This calculation was carried out for the truth model, and the upper left 16x16 portion of the resulting $\underline{Q}_d$ matrix was retained for the filter. The results of this computation are listed in Table IV (parts a and b).

As mentioned in Chapter II, this derivation does not account for noises which are removed when states are removed from the truth model. Thus, pseudo-noises have to be added to the computed results of Table IV. These pseudo-noises have to be tuned for best performance just as in the Type I filter.

Note that the only state dependent effect seen in Table IV is from the wander azimuth angle, α. This indicates that in a fixed azimuth system, a $\underline{Q}_d$ matrix derived in this manner would be a function only of Δt.

Tuning of the Type III filter should be easier than for the Type I because more of the driving noise has been derived analytically. This leaves only a few states for which pseudo-noises must be added to compensate for states removed from the truth model. The final values of these additional pseudo-noises are listed in Table IVc. Note

Table IV a.

$\underline{Q}_d$ Values for Filter Type III

Derived Diagonal Terms

$Q(3,3) = Q_{47} \cdot K_1^2 \cdot \Delta t^2$

$Q(4,4) = Q_{35} \cdot (\cos(\alpha) \cdot \Delta t)^2 + Q_{36} \cdot (\sin(\alpha) \cdot \Delta t)^2 + Q_{47} \cdot \Delta t^2$

$Q(5,5) = Q_{35} \cdot (\sin(\alpha) \cdot \Delta t)^2 + Q_{36} \cdot (\cos(\alpha) \cdot \Delta t)^2 + Q_{49} \cdot \Delta t^2$

$Q(6,6) = Q_{37} \cdot \Delta t^2 + Q_{47} \cdot K_2^2 \cdot \Delta t^2 + Q_{50} \cdot \Delta t^2$

$Q(7,7) = Q_{14} \cdot (\cos(\alpha) \cdot \Delta t)^2 + Q_{15} \cdot (\sin(\alpha) \cdot \Delta t)^2$

$Q(8,8) = Q_{14}\dot{\alpha}(\sin(\alpha) \cdot \Delta t)^2 + Q_{15} \cdot (\cos(\alpha) \cdot \Delta t)^2$

$Q(9,9) = Q_{16} \cdot \Delta t^2$

$Q(11,11) = Q_{47} \cdot K_3^2 \cdot \Delta t^2$

$Q(12,12) = 2 \cdot Q_{12} + Q_{52} \cdot \Delta t^2$

$Q(13,13) = Q_{52} \cdot \Delta t^2$

$Q(14,14) = 2 \cdot Q_{14}$

$Q(15,15) = 2 \cdot Q_{15}$

$Q(16,16) = 2 \cdot Q_{16}$

where

- α is the wander azimuth angle of the INS platform
- K_1, K_2, K_3 are gains for the third order vertical channel model.
- Q_n is the value of $\underline{Q}(n,n)$ from the truth model.

Table IV b.

$\underline{Q}_d$ Values for Filter Type III

Derived Off-Diagonal Terms

Q(3,6) = $K_1 \cdot K_2 \cdot Q_{47} \cdot \Delta t^2$

Q(3,11) = $-K_1 \cdot K_3 \cdot Q_{47} \cdot \Delta t^2$

Q(4,5) = $\sin(\alpha) \cdot \cos(\alpha) \cdot (Q_{35} - Q_{36}) \cdot \Delta t^2$

Q(6,11) = $-Q_{47} \cdot K_2 \cdot K_3 \cdot \Delta t^2$

Q(7,8) = $\sin(\alpha) \cdot \cos(\alpha) \cdot (Q_{14} - Q_{15}) \cdot \Delta t^2$

Q(7,14) = $Q_{14} \cdot \cos(\alpha) \cdot \Delta t$

Q(7,15) = $-Q_{15} \cdot \sin(\alpha) \cdot \Delta t$

Q(8,14) = $Q_{14} \cdot \sin(\alpha) \cdot \Delta t$

Q(8,15) = $Q_{15} \cdot \cos(\alpha) \cdot \Delta t$

Q(9,16) = $Q_{16} \cdot \Delta t$

Q(12,13) = $Q_{52} \cdot \Delta t^2$

$\underline{Q}_d$ is symmetric, so the symmetric terms must also be added, E.g., Q(6,3)=Q(3,6)

Table IV c.

$\underline{Q}_d$ Values for Filter Type III

Added Pseudo-Noises

$Q(1,1) = 4\times10^{-15}$

$Q(2,2) = 4\times10^{-15}$

$Q(3,3) =$ Above term + 600

$Q(7,7) =$ Above term + 5×10^{-8}

$Q(8,8) =$ Above term + 5×10^{-8}

$Q(9,9) =$ Above term + 5×10^{-8}

$Q(10,10) = 8\times10^{-8}$

$Q(11,11) =$ Above term + 1×10^{-3}

$Q(12,12) =$ Above term + 700

To obtain the final $\underline{Q}_d$ matrix, add the pseudo-noises in Table IVc to the computed terms in IVa and IVb and multiply by $\Delta t/2$.

that only nine pseudo-noises had to be added in the Type I filter.

Type IV State Noise Computation

This filter is based on the work of Widnall (Ref 5, Ref 6). He derived values for state dependent, on-diagonal pseudo-noises based on an analysis of the true system equations (equation (3)). Rather than simply discard the excess states in the truth model when designing the filter model, he lumped them all under the category of driving noise. Then he determined the appropriate strength for a single white Gaussian noise to simulate the result of these separate noise contributions. For several of the states, this had to be approximated in order to assure that the noise contributions added in each integration step would add up to the appropriate amount of noise for a complete maneuver. For other states, the errors could come from several sources, so the error source with the largest covariance is selected. This technique was used to generate $\underset{\sim}{Q}_d$ relations for states 4 through 11, and these are listed in Table V a.

The remaining on-diagonal terms were filled in with terms computed for the Type III filter $\underset{\sim}{Q}_d$ matrix. An examination of this matrix (Table IV) shows that many of the off-diagonal terms can be computed as the product of the square roots of the corresponding diagonal elements and a fixed correlation factor. For the others, the correlation factor varies as a function of the wander azimuth

Table V a.

Q_d Values for Filter Type IV

Derived On-Diagonal Terms

$Q(4,4) = 2 \cdot V \cdot F_s \cdot (\sigma_{M1})^2/\Delta t + 2 \cdot \tau_{GE} \cdot (\sigma_{GE})^2$

$Q(5,5) = 2 \cdot V \cdot F_s \cdot (\sigma_{M1})^2/\Delta t + 2 \cdot \tau_{GN} \cdot (\sigma_{GN})^2$

$Q(6,6) = 2 \cdot V \cdot F_s \cdot (\sigma_{M1})^2$

$Q(7,7) = 2 \cdot V \cdot F_s \cdot (\sigma_{M2})^2$

$Q(8,8) = Q(7.7)$

$Q(9,9) = Q(7,7)$

$Q(10,10) = 2 \cdot \beta_{17} \cdot (\sigma_{17})^2 + 2 \cdot \beta_{16} (\sigma_{16})^2/h^2 + 2 \cdot |h| \cdot (\sigma_{16})^2/(h^3 \Delta t)$

$Q(11,11) = (\sigma_{ABZ})^2 + 2 \cdot (\sigma_{GZ})^2/\tau_{GZ}$

where:

V = Path velocity at beginning of interval (ft/sec).

F_s = Magnitude of the specific force vector in the horizontal plane (ft/sec).

σ_{M1} = Largest standard deviation among 3 accelerometer scale factor errors and 6 gyro input axis misalignment angles (8.73×10^{-10} rad.).

σ_{M2} = Standard deviation of gyro drift coefficient (0.3°/hr/g)

$\tau_{GE}, \tau_{GN}, \tau_{GZ}$ = Gravity anomaly correlation times. Computed as $\tau_x = D_x/V$.

D_{GE}, D_{GN}, D_{GZ} = Gravity deflection correlation distances (values: $D_{GE} = D_{GN} = 60{,}800$ ft.), $D_{GZ} = 364{,}800$ ft).

$\sigma_{GE}, \sigma_{GN}, \sigma_{GZ}$ = Standard deviations of gravity deflections (values: 8.372×10^{-4}, 5.47×10^{-4}, and 1.127×10^{-3} ft/sec^2).

β_{16} = 1/correlation time for barometric altimeter bias state due to weather effects (first order Markov model). Computed as V/D_{ALT}; $D_{ALT} = 250$ n.m.

σ_{16} = Standard deviation of altimeter bias (500 feet)

β_{17} = 1/correlation time for altimeter scale factor error (1/7200 sec)

σ_{17} = Standard deviation of altimeter scale factor error (.03)

Δt = Integration step size, seconds

h = Altitude, feet

Table V b.

Q_d Values for Filter Type IV

Terms Copied from Type III Filter

$Q(3,3) = Q_{47} \cdot K_1^2 \cdot \Delta t^2/2$

$Q(12,12) = Q_{12} + Q_{52} \cdot \Delta t^2/2$

$Q(13,13) = Q_{52} \cdot \Delta t^2/2$

$Q(14,14) = Q_{14}$

$Q(15,15) = Q_{15}$

$Q(16,16) = Q_{16}$

where terms are as defined in Table IV a.

Pseudo-Noises Added for Tuning

$Q(1,1) = 2 \times 10^{-15}$ (rad^2/sec)

$Q(2,2) = 2 \times 10^{-15}$ (rad^2/sec)

$Q(3,3) =$ above term+300 (ft^2/sec)

$Q(12,12) =$ above term + 350 (ft^2/sec)

Table V c

$\underline{Q}_d$ Values for Filter Type IV

Terms Computed as Correlations of Diagonal Terms

$Q(3,6) = \sqrt{Q(3,3)} \times \sqrt{Q(6,6)} \times [1]$

$Q(3,11) = \sqrt{Q(3,3)} \times \sqrt{Q(11,11)} \times [-1]$

$Q(4,5) = 0$

$Q(6,11) = \sqrt{Q(6,6)} \times \sqrt{Q(11,11)} \times [1]$

$Q(7,8) = 0$

$Q(7,14) = \sqrt{Q(7,7)} \times \sqrt{Q(14,14)} \times [\cos(\alpha)/\sqrt{2}]$

$Q(7,15) = \sqrt{Q(7,7)} \times \sqrt{Q(15,15)} \times [-\sin(\alpha)/\sqrt{2}]$

$Q(8,14) = \sqrt{Q(8,8)} \times \sqrt{Q(14,14)} \times [\sin(\alpha)/\sqrt{2}]$

$Q(8,15) = \sqrt{Q(8,8)} \times \sqrt{Q(15,15)} \times [\cos(\alpha)/\sqrt{2}]$

$Q(9,16) = \sqrt{Q(9,9)} \times \sqrt{Q(16,16)} \times [1/\sqrt{2}]$

$Q(12,13) = \sqrt{Q(12,12)} \times \sqrt{Q(13,13)} \times [1]$

$\underline{Q}_d$ is symmetric, so, for example, $Q(6,3)=Q(3,6)$. To obtain $\underline{Q}_d$ from the given terms, multiply each term by Δt. (Note: the correlation factors given here depend on the fact that the truth model driving noise terms Q14 and Q15 are equal. If they were not equal, the relationship between $Q_d(7,14)$, $Q_d(7,15)$, $Q_d(8,14)$, and $Q_d(8,15)$ would be more complicated, as can be seen from the equations in Table IV).

angle, α. For example,

$$Q_d(3,3) = Q_{47}\cdot K_1^2\cdot \Delta t^2\cdot(\Delta t/2)$$

$$Q_d(6,6) = Q_{47}\cdot K_2^2\cdot \Delta t^2\cdot(\Delta t/2)$$

$$Q_d(3,6) = Q_{47}\cdot K_1\cdot K_2\cdot \Delta t^2(\Delta t/2)$$

so that $Q_d(3,6) = \sqrt{Q_d(3,3)} \times \sqrt{Q_d(6,6)}$. In this case, correlation factor is 1. Similarly, $Q_d(8,15) = \sqrt{Q_d(6,$ $\sqrt{Q_d(15,15)} \times [\cos(\alpha)/\sqrt{2}]$, so the correlation factor $\cos(\alpha)/\sqrt{2}$.

For the Type IV $\underline{Q}_d$ matrix, the off-diagonal terms w generated from the on-diagonal values using these derived correlation factors. These on- and off-diagonal terms are listed in Table V c.

This type of filter still needs some tuning. The on-diagonal states which were taken from the Type III $\underline{Q}_d$ matrix must be tuned as before. The final tuned values are listed in Table V. The derivation of appropriate $\underline{Q}_d$ values has minimized the amount of tuning necessary for this filter.

Type II State Noise Computation

This $\underline{Q}_d$ matrix is an attempt to incorporate some of the effects of off-diagonal terms derived above with a minimum impact on the computational burden which the Kalman Fil imposes on the airborne computer. Thus constant terms, as in the

diagonal

angle, α. For example,

$$Q_d(3,3) = Q_{47} \cdot K_1^2 \cdot \Delta t^2 \cdot (\Delta t/2)$$
$$Q_d(6,6) = Q_{47} \cdot K_2^2 \cdot \Delta t^2 \cdot (\Delta t/2)$$
$$Q_d(3,6) = Q_{47} \cdot K_1 \cdot K_2 \cdot \Delta t^2 (\Delta t/2)$$

so that $Q_d(3,6) = \sqrt{Q_d(3,3)} \times \sqrt{Q_d(6,6)}$. In this case, the correlation factor is 1. Similarly, $Q_d(8,15) = \sqrt{Q_d(6,6)} \times \sqrt{Q_d(15,15)} \times [\cos(\alpha)/\sqrt{2}]$, so the correlation factor is $\cos(\alpha)/\sqrt{2}$.

For the Type IV $\underline{Q}_d$ matrix, the off-diagonal terms were generated from the on-diagonal values using these derived correlation factors. These on- and off-diagonal terms are listed in Table V c.

This type of filter still needs some tuning. The on-diagonal states which were taken from the Type III $\underline{Q}_d$ matrix must be tuned as before. The final tuned values are listed in Table V. The derivation of appropriate $\underline{Q}_d$ values has minimized the amount of tuning necessary for this filter.

Type II State Noise Computation

This $\underline{Q}_d$ matrix is an attempt to incorporate some of the effects of off-diagonal terms derived above with a minimum impact on the computational burden which the Kalman Filer imposes on the airborne computer. Thus, it includes only constant terms, as in the Type I $\underline{Q}_d$ matrix, but adds off-diagonal terms. This change could be implemented in an operational filter with only a few extra additions for each propagation and a few extra words of storage.

In the derivation of the Type III $\underline{Q}_d$ matrix, 11 distinct off-diagonal terms found (there are 22 off-diagonal terms, but $\underline{Q}_d$ is symmetric, so only 11 must be stored). As can be seen from Table V, two of these terms have correlation factors of zero, and two have correlation factors of sin (α). Since these terms must be constant, it is necessary to select a reasonable value for α. In many applications, α is set to zero at the start of a flight, and stays fairly close to zero, so a value of zero was chosen for α. This also represents the effective value of α for a fixed azimuth INS. This causes the correlation factors sin (α) to go to zero also, so two more correlated terms drop out.

This leaves a total of 23 parameters in the $\underline{Q}_d$ matrix: 16 diagonal terms and 7 distinct off-diagonal terms. To minimize the impact of these extra terms on filter tuning, the off-diagonal terms were not individually tuned. The diagonal terms were varied in the tuning process, and then the off-diagonal terms were computed using the correlation factors derived from the Type III $\underline{Q}_d$ matrix (these are the same correlation factors used for the Type IV matrix). The final values of $\underline{Q}_d$ for filter Type II are listed in Table VI.

Table VI a.

$\underline{Q}_d$ Values for Filter Type II

Diagonal Terms

Q(1,1) = 2×10^{-15}

Q(2,2) = 2×10^{-15}

Q(3,3) = 1000

Q(4,4) = .01

Q(5,5) = .01

Q(6,6) = .01

Q(7,7) = 2.5×10^{-10}

Q(8,8) = 2.5×10^{-10}

Q(9,9) = 2.5×10^{-10}

Q(10,10) = 4×10^{-8}

Q(11,11) = 5×10^{-4}

Q(12,12) = 400

Q(13,13) = 1×10^{-10}

Q(14,14) = 5.8×10^{-20}

Q(15,15) = 5.86×10^{-20}

Q(16,16) = 1.62×10^{-17}

Units for these numbers are the same as those in Table III.

Table VI b.

$\underline{Q}_d$ Values for Filter Type II

Off-Diagonal Terms

$$Q(3,6) = \sqrt{Q(3,3)} \times \sqrt{Q(6,6)} \times [1]$$

$$Q(3,11) = \sqrt{Q(3,3)} \times \sqrt{Q(11,11)} \times [-1]$$

$$Q(6,11) = \sqrt{Q(6,6)} \times \sqrt{Q(11,11)} \times [1]$$

$$Q(7,14) = \sqrt{Q(7,7)} \times \sqrt{Q(14,14)} \times [1/\sqrt{2}]$$

$$Q(3,15) = \sqrt{Q(8,8)} \times \sqrt{Q(15,15)} \times [1/\sqrt{2}]$$

$$Q(9,16) = \sqrt{Q(9,9)} \times \sqrt{Q(16,16)} \times [1/\sqrt{2}]$$

$$Q(12,13) = \sqrt{Q(12,12)} \times \sqrt{Q(13,13)} \times [1]$$

$\underline{Q}_d$ is symmetric, so for example, $Q(6,3)=Q(3,6)$. To obtain $\underline{Q}_d$, multiply each of the above terms by Δt.

The method used for testing the Kalman Filters based on each of these proposed state noise covariance matrices is described in the next chapter.

IV. Evaluation of Proposed Filters

Introduction

Previous chapters of this report introduced the Kalman Filter, and discussed some of the approximations and simplification techniques which are needed to make it practical. Particular attention was focused on the need for an accurate computation of the driving noise covariance matrix, $\underline{Q}_d$. The defining equation was shown and the usual approximation was described. Then some alternate forms of $\underline{Q}_d$ calculations were depicted.

In order to evaluate these alternative computations, some method of assessing the performance of the resulting Kalman Filters must be used. The best test of a filter is to implement it in an airborne navigation system and gather actual flight data. The cost of such testing is very high, so it is generally limited to providing final performance verification of filters which have already undergone extensive testing. Thus, this study is limited to the use of computer simulations for performance analysis.

This chapter describes the testing process used for this research. It discusses the nature of the computer simulation used and the reasons for the selection of that type of simulation. It describes the flight profile over which the simulation was run, and the different parameters which were used in testing. The test results and analysis and discussion of these results will be presented in the next chapter.

Covariance Analysis

There are two basic types of computer simulations available for Kalman Filter testing. They are the covariance analysis and the Monte Carlo analysis. Both depend on the use of both the truth model and the filter itself to analyze the performance of the Kalman Filter.

In a Monte Carlo analysis, the simulation run uses the Kalman Filter just as it would be used in flight. The computer uses a random number generator to generate the driving noise for the truth model and presents appropriately noise-corrupted external measurements, $\underline{z}$, to the filter. The filter generates a state estimate, $\hat{\underline{x}}$, based on these measurements. Then these estimates are compared with the true state values generated by the truth model, and errors are computed. In order to have some confidence in the statistical accuracy of these errors, several runs must be made over the same profile to get a statistically significant sample. Then the sample mean and the covariance of the errors can be computed.

The Monte Carlo analysis provides an accurate test of a linear Kalman Filter, or of nonlinear variants such as the Extended Kalman Filter. However, for a long flight profile, it takes an excessive amount of computer time to generate statistically significant results. A faster simulation is possible for some problems in the form of the covariance analysis.

The basis for the covariance analysis technique comes from Kalman Filter theory and linear system theory. Under the assumptions that the truth model is a linear system driven by white Gaussian noise, and the Kalman Filter is based directly on the truth model, it can be shown that the covariance which the filter computes for the state estimate is the same as the covariance which the filter commits in estimating the state. It can be seen from equations (12), (13), and (15) that this covariance, $\underline{P}$, can be computed for all time without knowledge of the values of the measurements (note that $\underline{z}$ does not appear in equations (12), (13) and (15)). Thus, the covariance analysis computes the covariance of the error in the state estimates directly. There is not need for random number generators to supply driving noises or for a large number of test runs to generate valid statistics.

Practical Kalman Filters are based on a simplified, reduced order model of the system, so the assumption that the filter model accurately describes the behavior of the system is not valid. However, in an off-line simulation, both the truth model and the filter model can be used. In this way, the actual errors committed by the filter in estimating the state can be evaluated (Ref 7).

Let the subscript "s" denote the system (truth) model and "f" denote the filter model components. Then the error committed by the filter is simply the difference between the true state, $\underline{x}_s$, and the filter state estimate, $\hat{\underline{x}}_f$.

Since these states are of different dimension, this is written:

$$\underline{e}(t) = \underline{x}_s(t) - \underline{T}\hat{\underline{x}}_f\,(t) \tag{21}$$

where $\underline{e}$ is an error vector and $\underline{T}$ is a transformation matrix from the true state to the filter state. A common for $\underline{T}$ is $[\underline{I}:\underline{0}]^T$, indicating that the filter contains the first n states from the truth model. The equations for propagating and updating the covariance of this error, $\underline{P}_e$, can be derived just as equations (12), (13), and (15) were derived for the covariance estimate in the filter, $\underline{P}_f$.

For the special case in which the models contain only error states and an impulsive control is available to reset all of the estimated errors to zero for each measurement, these covariance equations have an especially convenient form. In this case, the error covariance, $\underline{P}_e$, satisfies the same equation as the system covariance, P_s. The time propagation equation for $\underline{P}_e$ is equation (12) or (17). In practice, the gain matrix computed by the filter, $\underline{K}_f$, will be used to update the state estimate, so it must be used to update the truth model covariance. Thus, for a covariance analysis, the Kalman Filter being tested must be used to compute the gain matrix through equations (12), (13), and (15). Since $\underline{K}_f$ is not the same as the theoretically optimal gain which a filter based on the truth model would compute, the update equation for $\underline{P}_e$ must be modified from the form of equation (15). The result is:

$$\underline{P}_e^+ = (\underline{I}-\underline{T}\underline{K}_f\underline{H}_s)\underline{P}_e^- - (\underline{I}-\underline{T}\underline{K}_f\underline{H}_s)^T+\underline{T}\underline{K}_f\underline{R}_s\underline{K}_f^T\underline{T}^T \qquad (22)$$

Thus, by using both the truth model and the filter model, the covariance of the true errors committed by the filter in estimating the state can be computed through the use of equations (12), (13), (15), and (22). Note that there is still no dependence on the specific measurements $\underline{z}(t_i)$ which the filter would receive. Therefore, it is still possible to compute directly the covariance of the error in the state estimate in a single simulation run.

The covariance analysis has several limitations. It computes only the covariance of the errors, ignoring their mean value. By the nature of its equations, sign errors in the filter model matrices can go undetected. The most serious limitation is that a covariance analysis is theoretically correct only for linear Kalman Filter and truth models.

As described in Chapter II, the filter in this study is not strictly linear, but rather is an Extended Kalman Filter, which is linearized about a nominal path, $\underline{x}_n$. In actual use, a new nominal path would be computed after each measurement incorporation. However, in a covariance analysis, there are no measurements available. Information about some nominal path is supplied to the filter by the covariance analysis program, but there is no way to assure that the filter estimates would actually stay near this nominal path.

In a covariance analysis, the state dependent terms in the model are always based on the nominal state, $\underline{x}_n$. In on-line use, these terms must be based on the estimated state, $\hat{\underline{x}}$. The difference between these two values of $\underline{x}$ can cause the $\underline{F}(t)$ matrix, as evaluated in equation (2), to vary significantly between the covariance analysis tests and on-line application. Thus, a filter that works well in a covariance analysis may perform less well in actual use.

For this reason, a covariance analysis usually plays a limited role in the development of a Kalman Filter. It is usually used for tuning because of its faster run time. Once this tuning is completed, the filter is tested in a more accurate way (Monte Carlo analysis and/or flight testing) to assure proper performance. The covariance analysis results are viewed as a limited, initial indication of performance.

In this study, a covariance analysis is used because of computer time considerations. Thus, the results presented are limited by the problems noted above. However, the intent of this study is to evaluate the *relative* performance of the proposed Kalman Filters. Since the filters differ only in the computation of the driving noise covariance matrix, $\underline{Q}_d$, nonlinearities and other unseen effects in the covariance analysis are likely to be approximately the same for all of them. Thus, the covariance analysis provides a valid method for the required performance analysis.

The covariance analyses for this study were run using a computer program supplied by the Air Force Avionics Laboratory known as the General Covariance Analysis Program (GCAP) (Ref 7). This program provides utility subroutines for storing and manipulating the truth model and filter model matrices, covariance matrices, Kalman gain matrix, and so on. The user supplies subroutines to read in his input data, compute a nominal path, and compute the components of his truth model and filter model matrices.

The GCAP program propagates the covariance in time through the use of a continuous time differential equation of a different form from equation (12):

$$\dot{\underline{P}}(t) = \underline{F}(t)\underline{P}(t) + \underline{P}(t)\underline{F}^T(t) + \underline{G}(t)\underline{Q}(t)\underline{G}^T(t) \qquad (23)$$

This equation has the same solution as equation (12), so the results of this equation are valid for the current study. The GCAP program integrates $\dot{\underline{P}}(t)$ given in equation (23) using a fourth-order, Runge-Kutta integration routine. This routine computes new values of $\underline{F}(t)$ and $\underline{G}(t)\underline{Q}(t)\underline{G}^T(t)$ at the beginning, middle, and end of each integration step. Thus, this routine provides a more accurate numerical propagation of the covariance than equation (12) for the same integration step size.

Equation (23) causes a problem for this study. It requires a value for $\underline{Q}(t)$, but the study is testing forms for the more commonly used $\underline{Q}_d(t)$. Thus, the propagation of the filter covariance is separated into two parts. The homogeneous part is computed using equation (23) with $\underline{Q}(t)$ set to $\underline{0}$. Then the effect of the driving noise, $\underline{Q}_d$, is added to this result. This calculation is more representative of the type of calculation used in on-line Kalman Filter applications than a direct integration of equation (23). A direct evaluation of (23) is used for the truth model to provide greater accuracy.

The GCAP program provides the needed calls to a user-supplied subroutine (TRAJ) to generate the nominal path, $\underline{x}_n$. Then it calls another subroutine (FLTMAT or SYSMAT) to compute the values of the filter or truth model matrix elements for that value of $\underline{x}_n$. These matrices are then used for covariance propagation and updating. The basic sequence is as follows:

1. Initialize matrices.
2. Compute the nominal state, $\underline{x}_n$.
3. Compute linearized filter matrices, $\underline{F}_f$ and $\underline{Q}_{df}$.
4. Propagate the filter covariance matrix, $\underline{P}_f$, one integration step forward in time (i.e., integrate equation (23) with $\underline{Q}(t)=\underline{0}$ for one integration step, then add $\underline{Q}_{df}(t_i)$).
5. Repeat steps 2-4 until it is time for a measurement update.
6. Compute matrices $\underline{H}_f$ and $\underline{R}_f$.
7. Compute a Kalman gain matrix, $\underline{K}_f$, using equation (13). Use this gain matrix to update $\underline{P}_f$.

8. Repeat steps 2-6 using the truth model matrices, $\underline{F}_s$, $\underline{Q}_s$, $\underline{H}_s$, and $\underline{R}_s$, and the true error covariance matrix $\underline{P}_e$. (Use equation (23) directly in step 4.)

9. Use the Kalman gain matrix computed for the filter, $\underline{K}_f$, to update the true error covariance matrix, $\underline{P}_e$, using equation (22).

10. Repeat steps 2-9 until the specified stop time is reached.

The true error covariance matrix, $\underline{P}_e$, represents the covariance of the error committed by the filter in estimating the state. The program provides facilities for plotting both the filter covariance, $\underline{P}_f$, and the true error covariance, $\underline{P}_e$. This plotted output provides a convenient form for presenting test results, and in fact will be used for this purpose in the next chapter.

Test Parameters

As mentioned above, the GCAP program provides calls to a subroutine to generate a nominal path, $\underline{x}_n$. For this study, the subroutine simply reads values for $\underline{x}_n$ from a file generated by an external program. The external program used is called the Profile Generator Program, or PROFGEN (Ref 8). This program generates parameters such as position, velocity, attitude angles, and specific force vectors for an aircraft traversing a specified flight path. The user specifies the maneuvers he wishes the aircraft to perform: turns, pitches, climbs, dives, path accelerations, etc. In addition, he specifies certain aircraft performance factors, such as maximum roll rate. The program can simulate a wide range of maneuvers and gives very accurate computed values of the vehicle parameters.

Two factors in the study affected the type of flight profile selected. In order to show the benefits of state dependent terms in the noise matrix, a flight profile with large changes in dynamics such as velocity, specific force, and altitude was desired. This suggests a profile typical of a high performance aircraft, involving high and low flight, high-g turns, straight flight, and so on.

The second factor in profile selection is the integration step size. Small integration steps provide more accurate integration, but greatly increase the computational burden of the Kalman Filter. Normally, the designer must minimize this computational burden, so he would like to use as large a stepsize as possible. The stepsize must be small enough to allow a sampling frequency at least twice as high as the frequency corresponding to the fastest error state dynamics in the model, in order to avoid aliasing problems. From the viewpoint of computational burden, it is desirable to use a stepsize as near this limit as possible while maintaining sufficient accuracy.

From the derivation in Chapter III, it can be seen that the off-diagonal terms in the proposed $\underline{Q}_d$ matrices are proportional to Δt^2, while the on-diagonal terms and pseudonoises are proportional to Δt. Thus, if Δt is small, the off-diagonal terms will be significantly smaller than the on-diagonal terms, so the approximation of setting the off-diagonal terms to zero is justified. However, as Δt is increased, the magnitude of the off-diagonal terms increases, so for a large Δt, they become significant.

Furthermore, as the integration step size is reduced, more of these smaller integration steps must be computed. The process of integrating equation (23) and then adding $\underline{Q}_d$ is repeated several times between updates. Integration of equation (23) after $\underline{Q}_d$ has been added for the previous integration step causes off-diagonal terms to be computed for $\underline{P}$ even when $\underline{Q}_d$ is diagonal. Thus, to show the effects of including off-diagonal terms in the $\underline{Q}_d$ matrix most clearly, a large integration stepsize is desired.

The flight profile chosen for this study consists of a representative combat flight for an F-4 aircraft. The inputs to the PROFGEN program split the flight into segments. For each segment, the desired maneuver is specified. The maneuvers specified for the flight used for this study are listed in Table VII.

For this flight profile, an integration stepsize of 2 seconds was selected. This stepsize was found to give accurate tracking of the error state dynamics. Larger integration stepsizes were found to cause numerical problems in the GCAP program which caused the covariance matrices to have negative terms on the diagonals. This effect appears

Table VII a.

Flight Profile for Test Cases

Initial Conditions*:

Time = 4728 seconds
Velocity = 934.8 ft/sec.
Heading = 91.6 deg.
Pitch Angle = -2.04 deg.
Latitude = 42.43 deg.
Longitude = -72.25 deg. (72.25 deg. West)
Altitude = 17,743.4 feet

*NOTE: This profile starts in the middle of a longer flight, so the initial conditions are not round numbers. The times listed here will be seen on the plotted output, so the actual times are listed rather than elapsed times.

to be caused by the rapidly changing dynamics of the chosen flight profile.

Clearly, for a stepsize of 2 seconds, Δt^2 is not small compared to Δt, so this stepsize should provide a good test for the proposed alternative state noise covariance computations. To provide a comparison, the filters were also tested with an integration stepsize of 0.2 seconds. The filters would have to be retuned to give the best obtainable performance with this value of Δt. However, the computer cost for these runs was relatively high, so this was not done. Instead, the filters were retested using the $\underline{Q}_d$ matrix values listed in Table III to VI, solely to provide a basis of comparison for the longer integration stepsize runs.

The external measurements available to the filter are provided by a Global Positioning System (GPS) receiver. This system consists (when it becomes operational) of a set of 24 satellites evenly distributed in three orbital planes. The position of each satellite is known very accurately, and each broadcasts a very accurately timed sequence of pulses. By measuring the time delay between these pulses and pulses generated by an onboard clock which is synchronized with the satellite clocks, the position and velocity of the receiver can be measured very accurately. Errors in the onboard clock cause inaccuracies in the measurement in the form of a range bias and a range rate bias. To reduce these errors, extra "user clock" states to model the biases

Table VII b.

Flight Profile for Test Cases

Segment	Time	Action
1	4728-4733	Pitch down to -6.54 deg. and decelerate to 844.7 ft/sec.
2	4733-4769	Descending right 360 deg. 4.5 g turn.
3	4769-4805	Descending left 360 deg. 4.5 g turn.
4	4805-4815	Pitch up to +1.56 deg. (Altitude at end = 10586 ft.)
5	4815-4915	Sinusoidal heading changes, 14.4 deg/sec, climb to 12,886 ft.
6	4915-5095	Sinusoidal heading changes, 12 deg/sec, maximum yaw = 15 deg, climb to 17,025 ft.
7	5095-5110	Climbing 1 g 5 deg. right turn, climb to 17,370 ft.
8	5110-5135	Pitch to level flight, climb to 17,386 ft.
9	5135-5140	Straight flight. Ends at Lat. = 42.34 deg., Lon. = -71.21 deg., Alt = 17386 ft.
10	5140-5142.5	1 g, 2 deg. right turn.
11	5142.5-5168.5	Pitch down to -45 deg., descend to 9127 ft.
12	5168.5-5173	Continue descent to 6439 ft.
13	5173-5178	Descending 0.5 g 30 deg. right turn, descend to 3452 ft.

are added to the baro-inertial system model used in the filter.

For this study, it is assumed that four satellites are continuously in view in a fixed location with respect to the aircraft, and both range and range rate measurements are available from each satellite (this is somewhat different from the way the actual system will work, but the difference is not relevant to this study). To be realistic, the simulation should account for the fact that the satellites move. It should propagate them in time and optimally select a new set of them for measurements at regular intervals. However, the program logic to do this is fairly complex and time consuming, and has no bearing on the issue of alternative $\underline{Q}_d$ evaluations. Therefore, the observation geometry matrix, $\underline{H}(t)$, was computed once, using satellite geometry for a randomly chosen time and location. Then this $\underline{H}$ matrix was assumed to be constant throughout all test runs.

Measurements were presented to the filter every 10 seconds. This update interval was chosen as a fairly representative value of the update intervals used in typical high accuracy, aided inertial navigation systems.

Kalman Filters for each of the proposed $\underline{Q}_d$ matrices were coded in the appropriate form for the GCAP program. Then a covariance analysis was performed for each one, using the parameters described in this chapter. The results of these tests are presented and discussed in the next chapter.

V Results and Discussion

Introduction

Previous chapters have introduced the problem to be evaluated in this study. The notation used and the mathematical formulation of the Kalman Filter were presented in Chapter II. The need for approximate computations in order to achieve practical on-line implementation was shown. In Chapter IV, some specific methods for evaluating the state noise covariance matrix, $\underline{Q}_d$, were derived. The methods used for testing Extended Kalman Filters based on these covariance matrices were described in Chapter IV.

This chapter presents the results of the testing for filters based on the four proposed $\underline{Q}_d$ matrices. The test results are shown in both graphical and tabular form. These results are discussed and analyzed, and suggestions for future studies are made.

Test Results

The results of the covariance analysis testing are presented in Figures 1 to 48. These plots present the square root of the estimation error covariances of the state variables listed. This "one-sigma" value of the error in the state estimate gives a description of the expected error, in that 68% of the errors will be less than or equal to this value. 99.8% of all errors will be less than or equal to three times this one-sigma value. All error states are assumed to have a mean value of zero.

The variables plotted are Longitude error, Latitude error, Altitude error, and East, North, and vertical components of velocity error. For each of the filters tested, two plots are shown for each variable. The first of these is labeled "FILTER" and depicts the covariance computed by the filter for each state estimate. The second plot is labeled "SYSTEM" and depicts the true error covariance computed by GCAP using the truth model system matrices and the Kalman gain matrix from the filter.

One set of plots is shown for each of the proposed filters. Recall that these filters differ only in the form of the state noise covariance matrix, $\underline{Q}_d$:

Type I - $\underline{Q}_d$ is a diagonal matrix, with constant terms.
Type II - $\underline{Q}_d$ is a full matrix, with constant terms.
Type III - $\underline{Q}_d$ is a full matrix, with varying terms derived by a trapezoidal integration of the defining integral (equation (22)).
Type IV - $\underline{Q}_d$ is a full matrix, with terms which vary as a function of the system state.

The plots shown were generated with an integration stepsize of 2.0 seconds. The scales for these plots are

Time - Mission time, seconds.
Longitude - Radians.
Latitude - Radians.
Altitude - Feet.
East Velocity - Feet/sec.
North Velocity - Feet/sec.

Vertical Velocity - Feet/sec.

Interpretation of Test Results

The two types of plots shown are to be interpreted differently. The plots labeled "FILTER" show, in root-mean-square (rms) form the covariance computed by the filter for its state estimate. The primary importance of this covariance information is its use in computing the Kalman gain matrix. If the computed covariance is too large, the gain will be larger than the optimal value, causing the filter to weight the measurement information too heavily. This causes the navigation system to track the noise in the measurements, and to fail to take advantage of all of the information available in its own state estimate. If the covariance is too small, the gain will be smaller than optimal, so the filter will weight the propagated state estimate too heavily, If the covariance becomes excessively small, the filter will ignore the external measurements. This is known as filter divergence, since it allows the errors in the navigation system to grow without bound. Thus, the "FILTER" covariance plots have only an indirect meaning, in that they influence the values of the actual errors committed by the filter.

The more directly meaningful plots are those labeled "SYSTEM". These depict the rms values of the true estimation errors, calculated by GCAP using the truth model system matrices and the Kalman gain matrix computed by the filter. The goal of Kalman Filter design and tuning is to

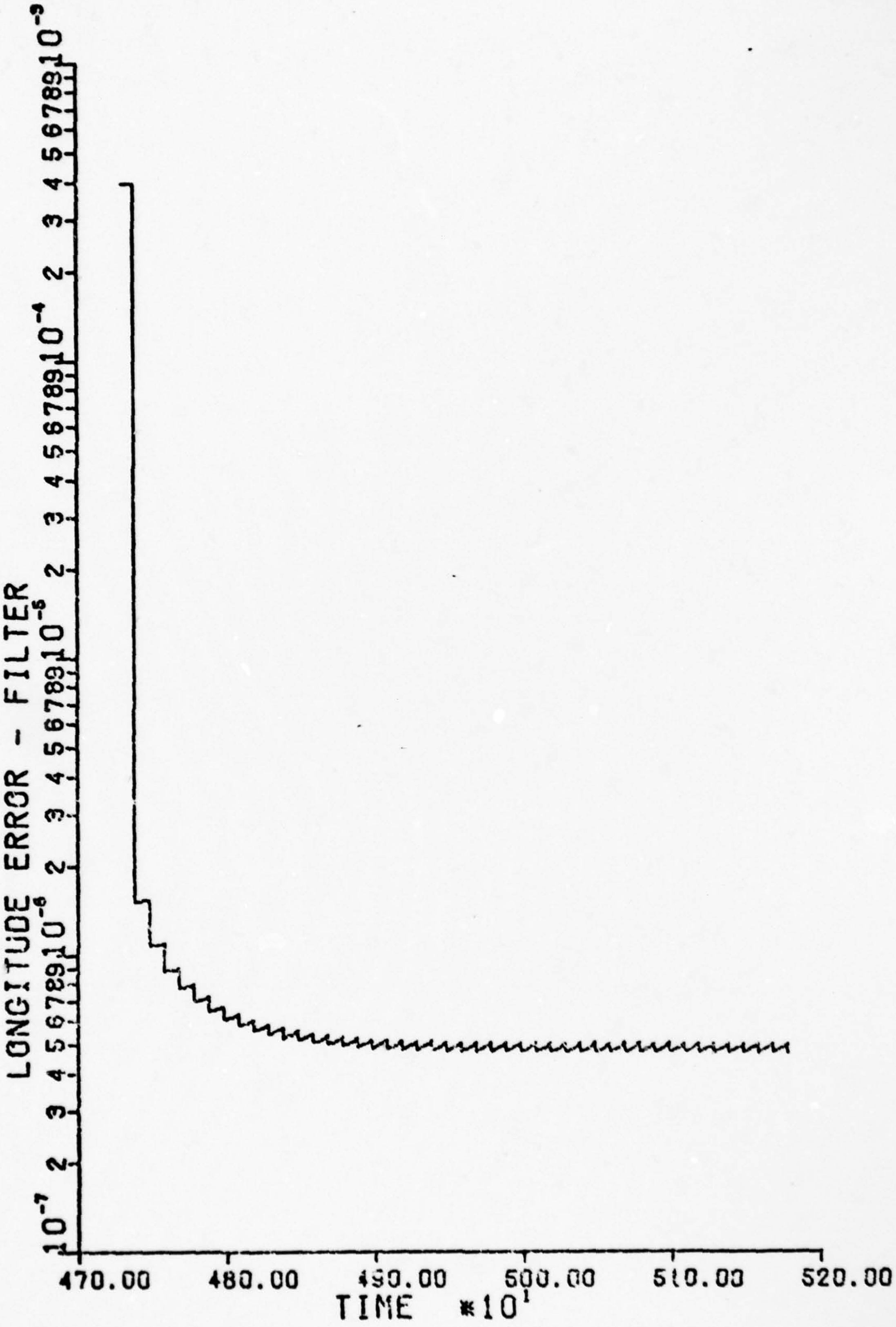


Figure 1. Filter I Estimated Longitude Error

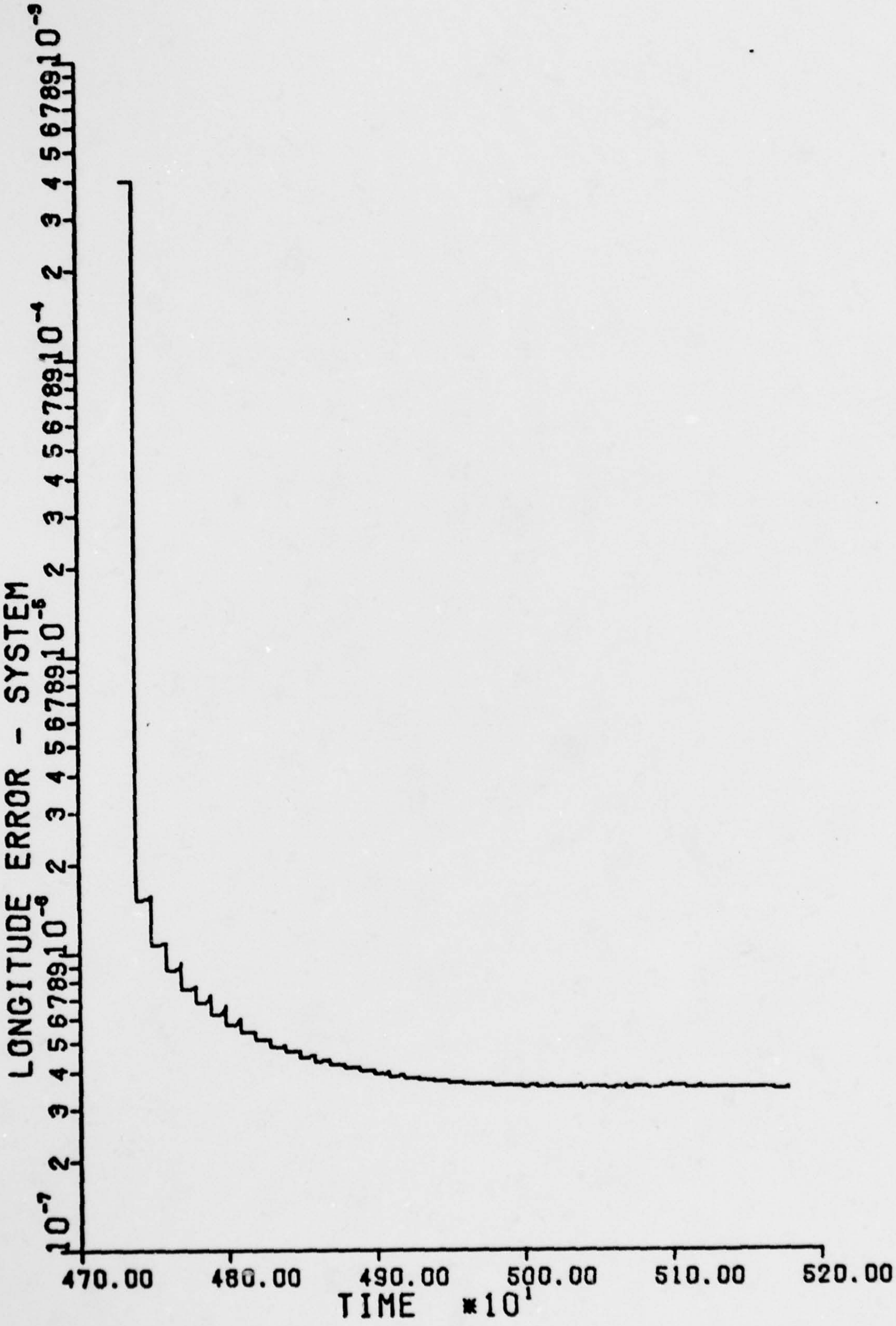


Figure 2. Filter I True Longitude Error

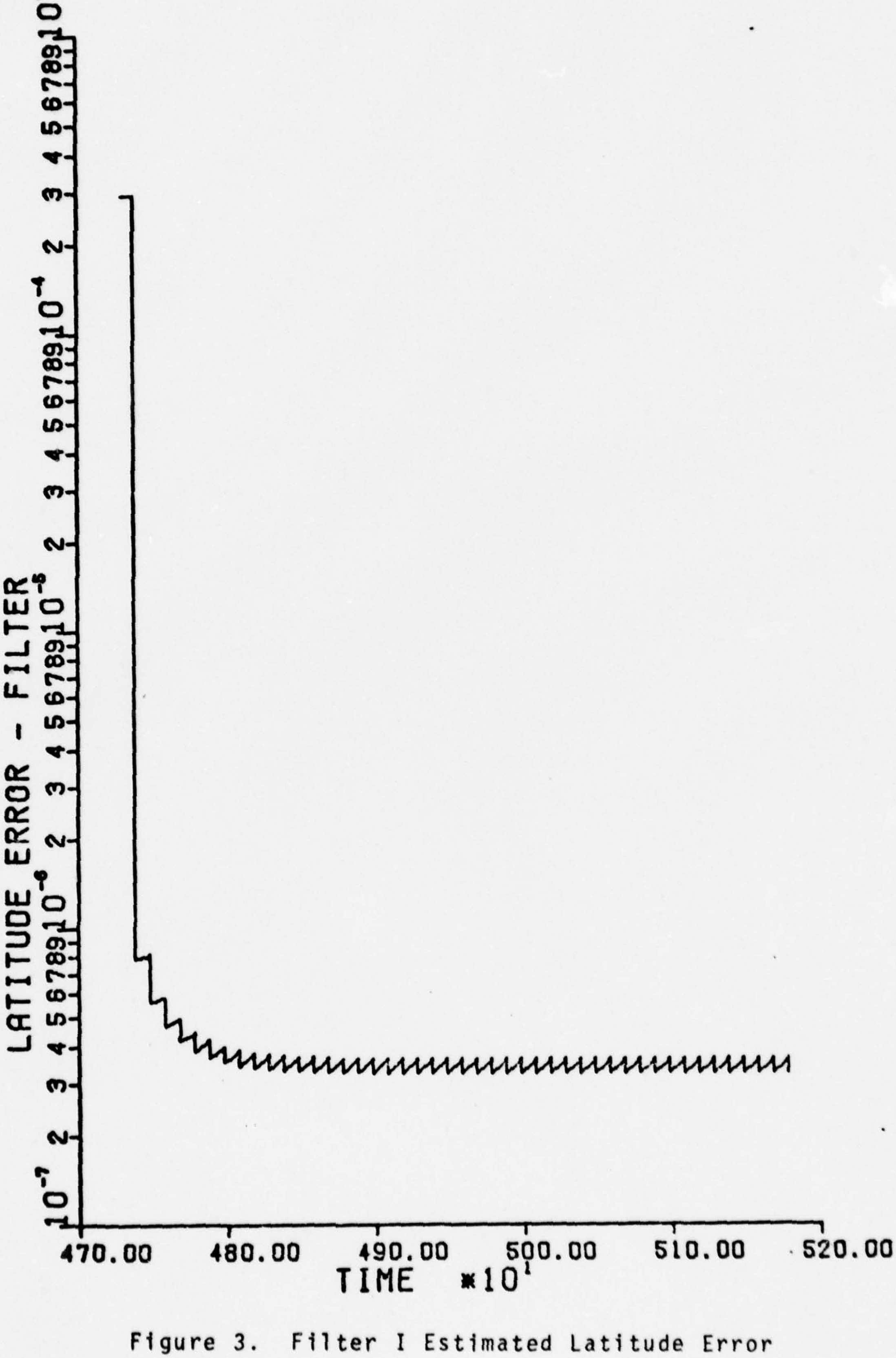


Figure 3. Filter I Estimated Latitude Error

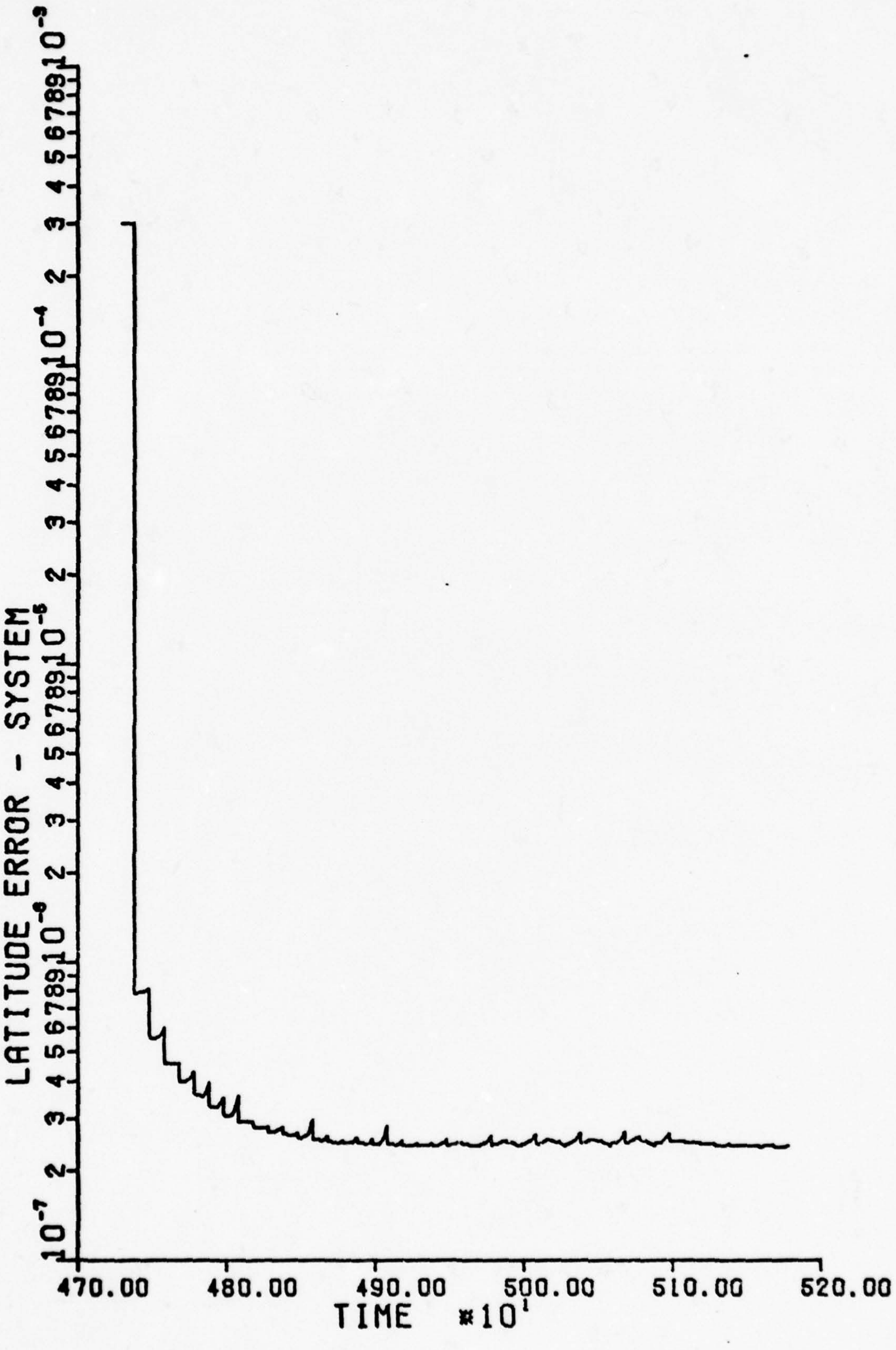


Figure 4. Filter I True Latitude Error

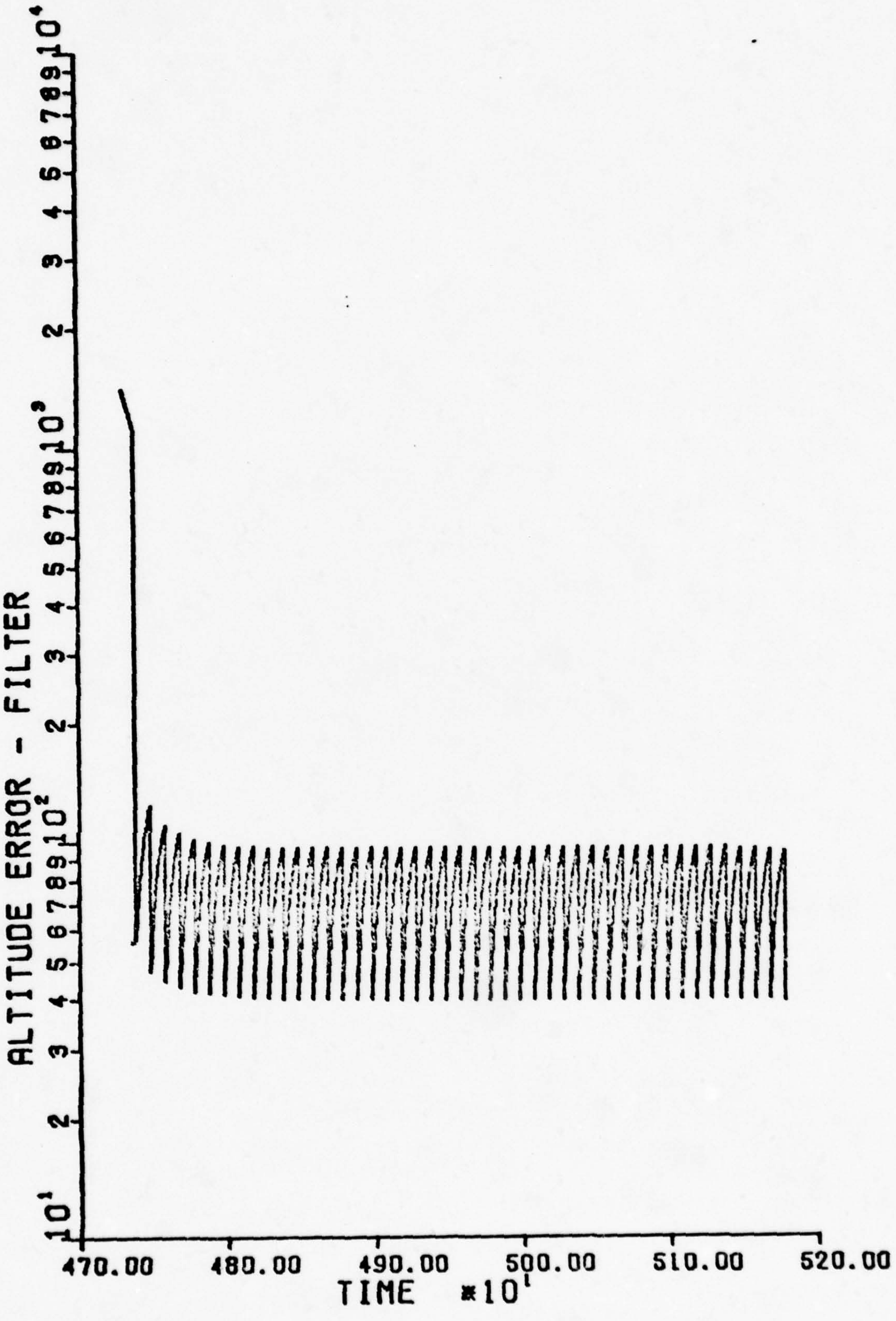


Figure 5. Filter I Estimated Altitude Error

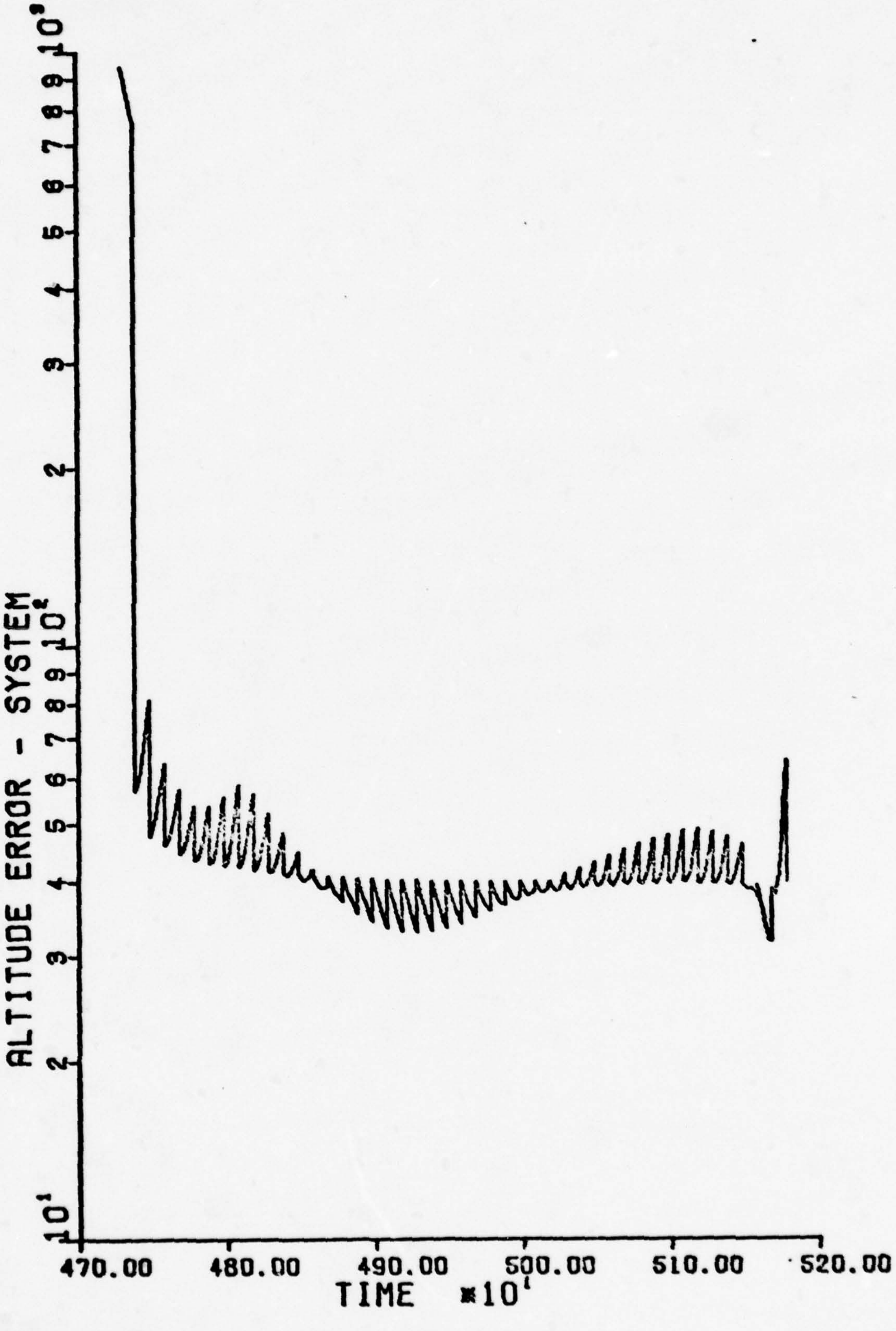


Figure 6. Filter I True Altitude Error

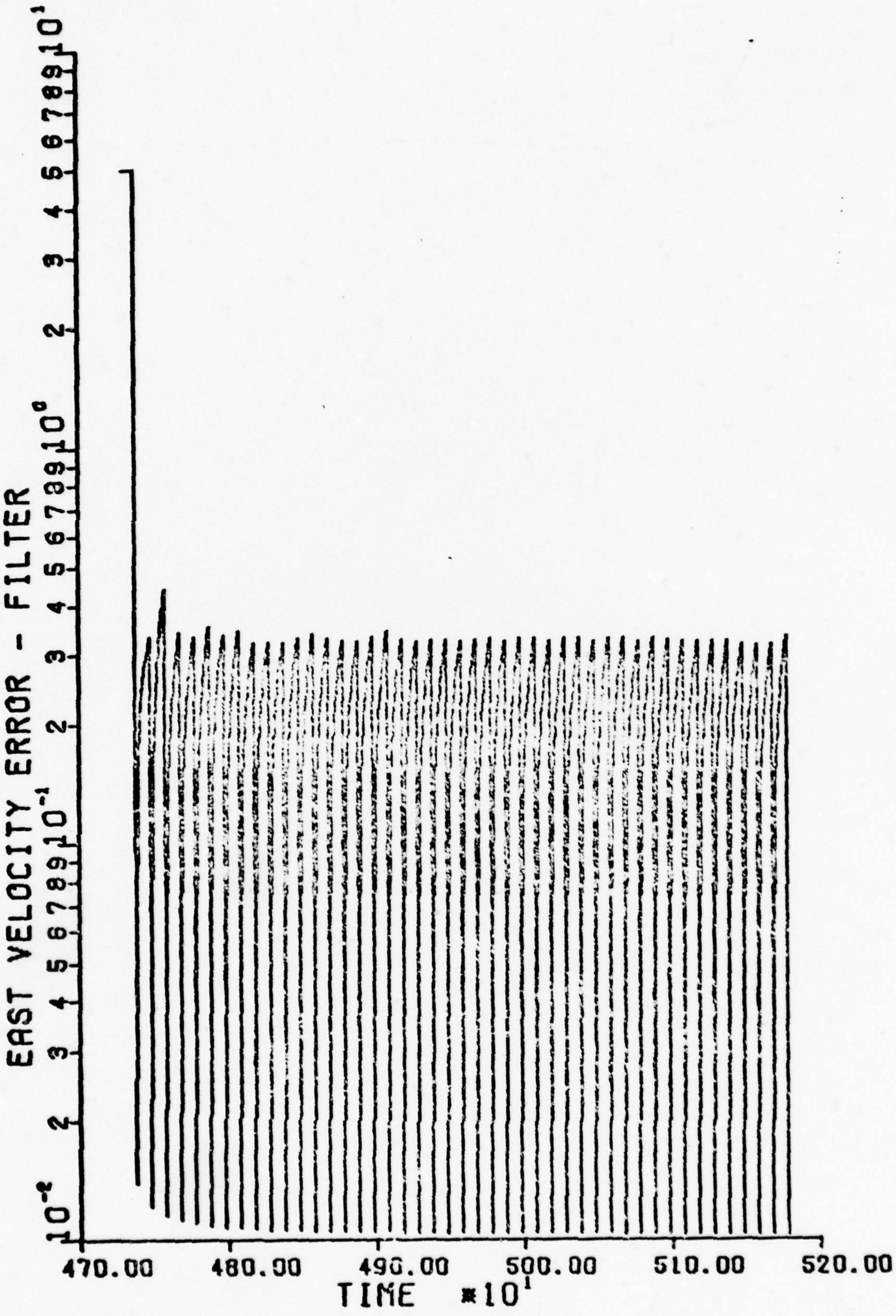


Figure 7. Estimated East Velocity Error

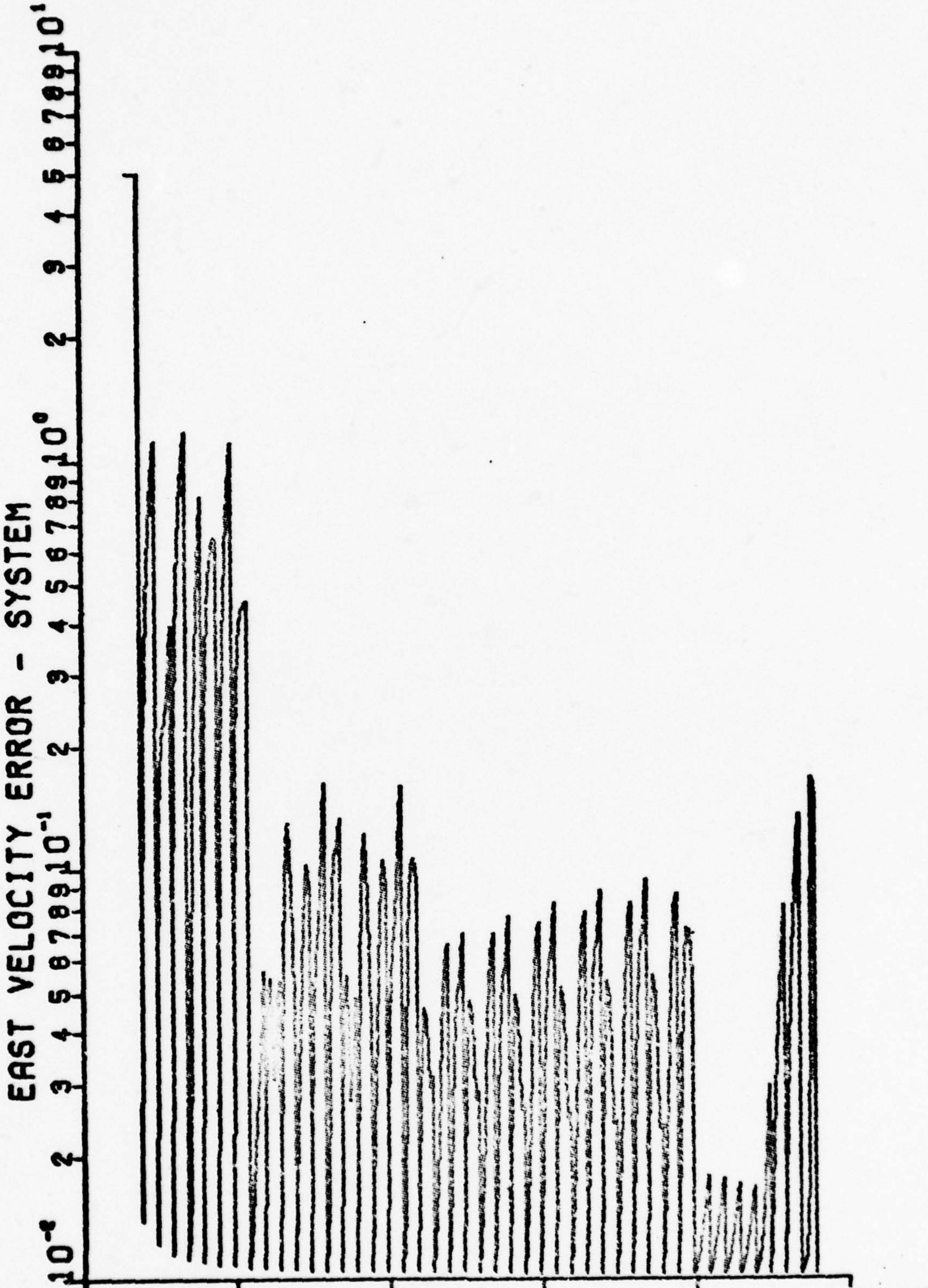


Figure 8. Filter I True East Velocity Error

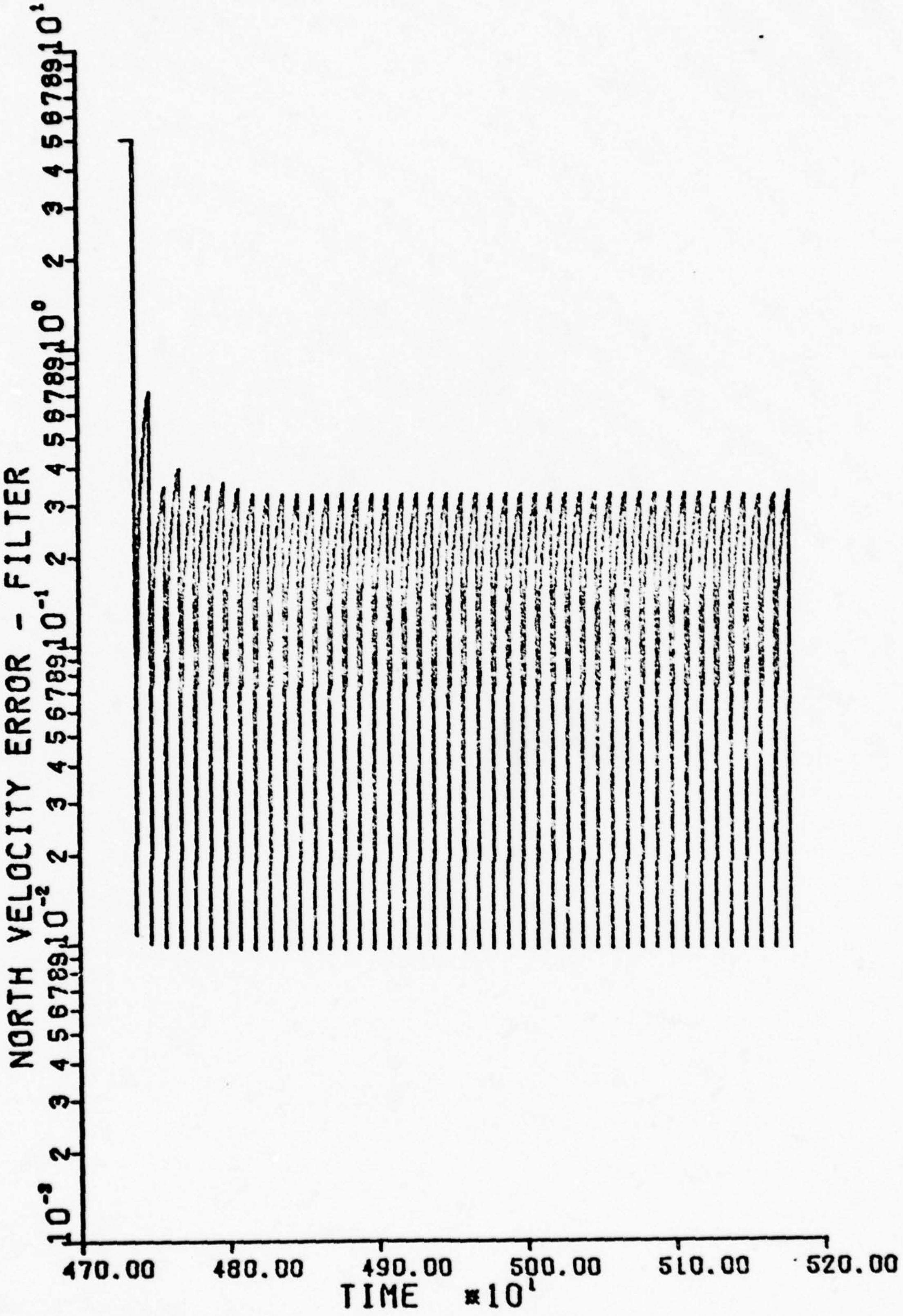


Figure 9. Filter I Estimated North Velocity Error

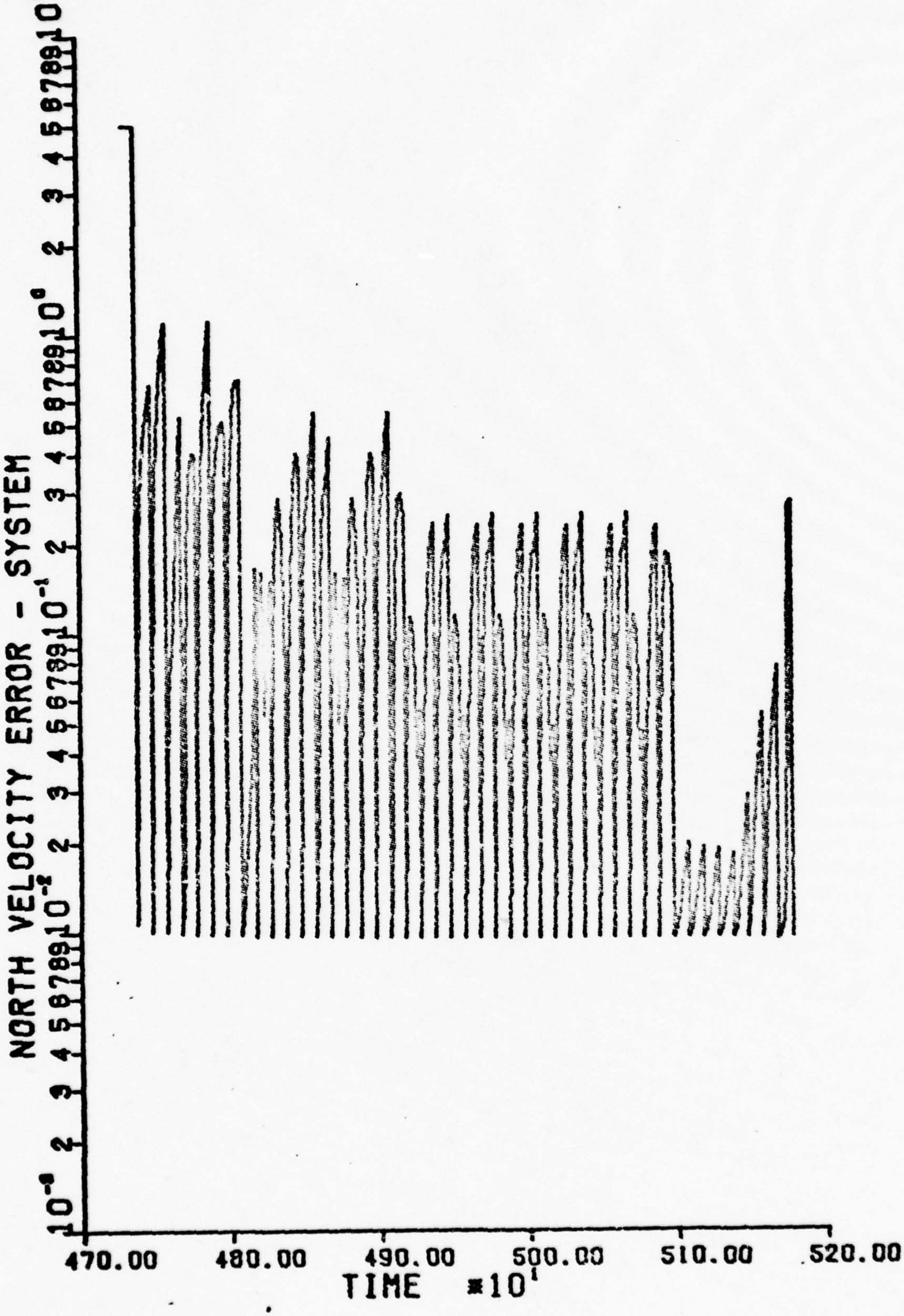


Figure 10. Filter I True North Velocity Error

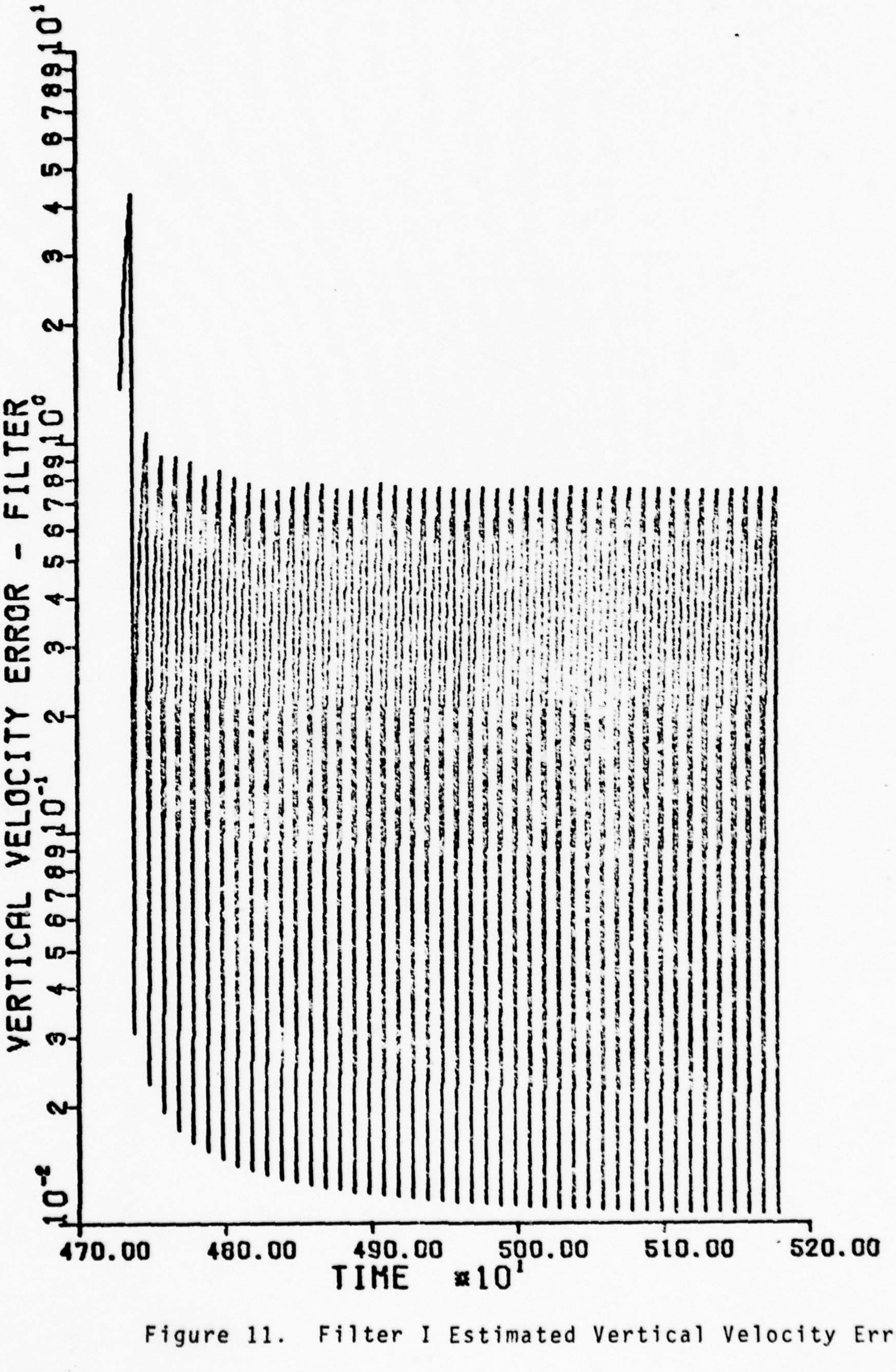


Figure 11. Filter I Estimated Vertical Velocity Error

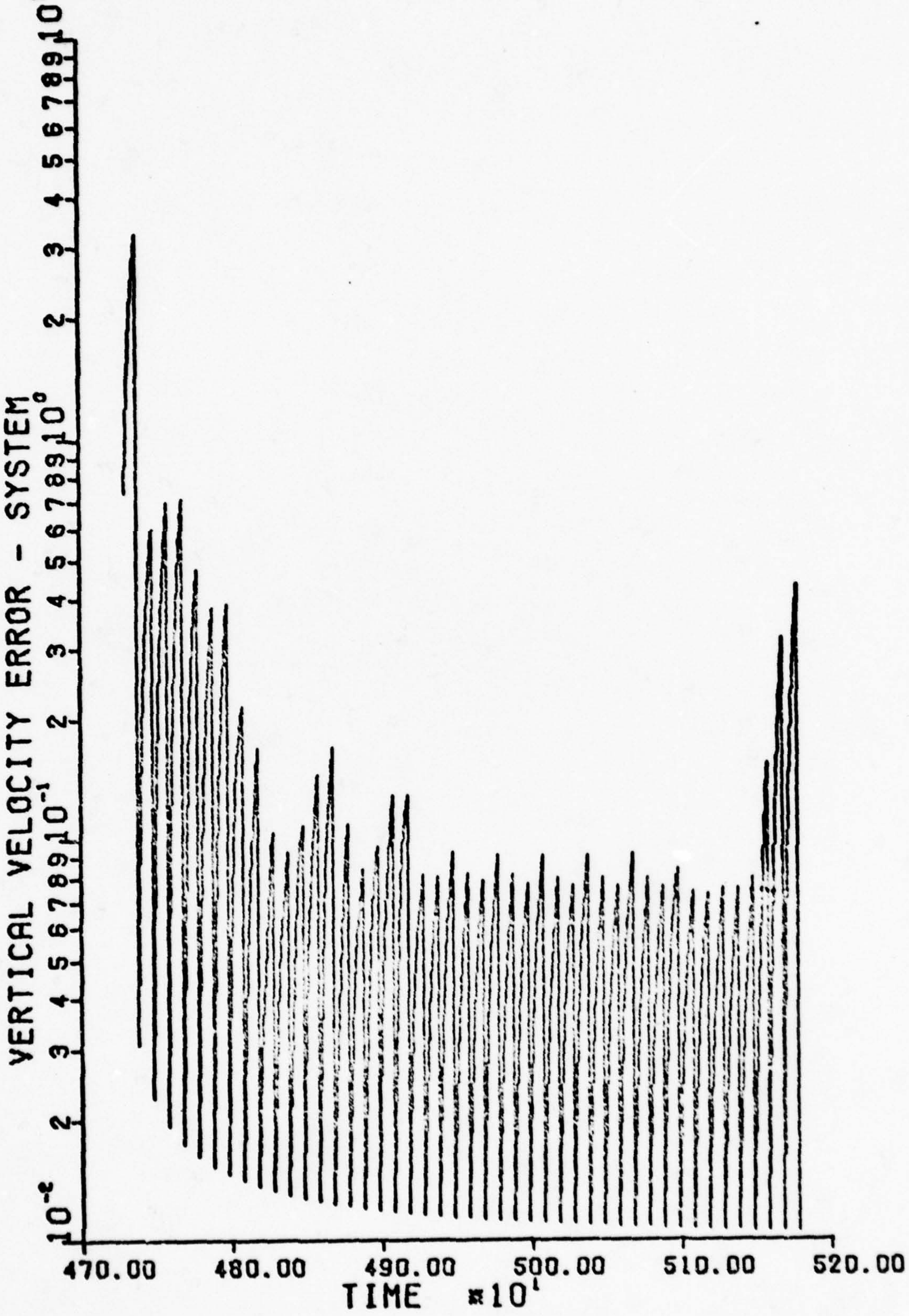


Figure 12. Filter I True Vertical Velocity Error

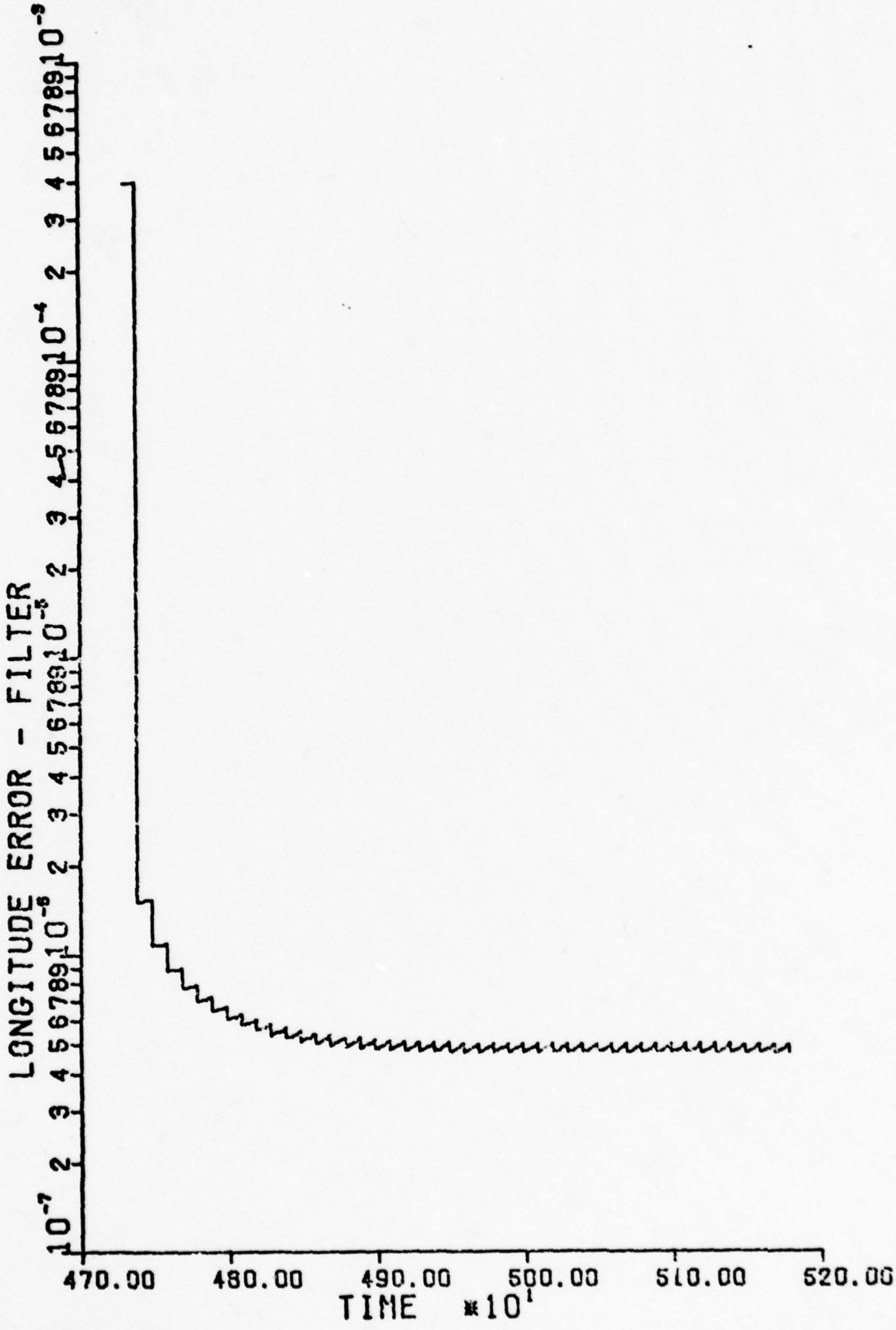


Figure 13. Filter II Estimated Longitude Error

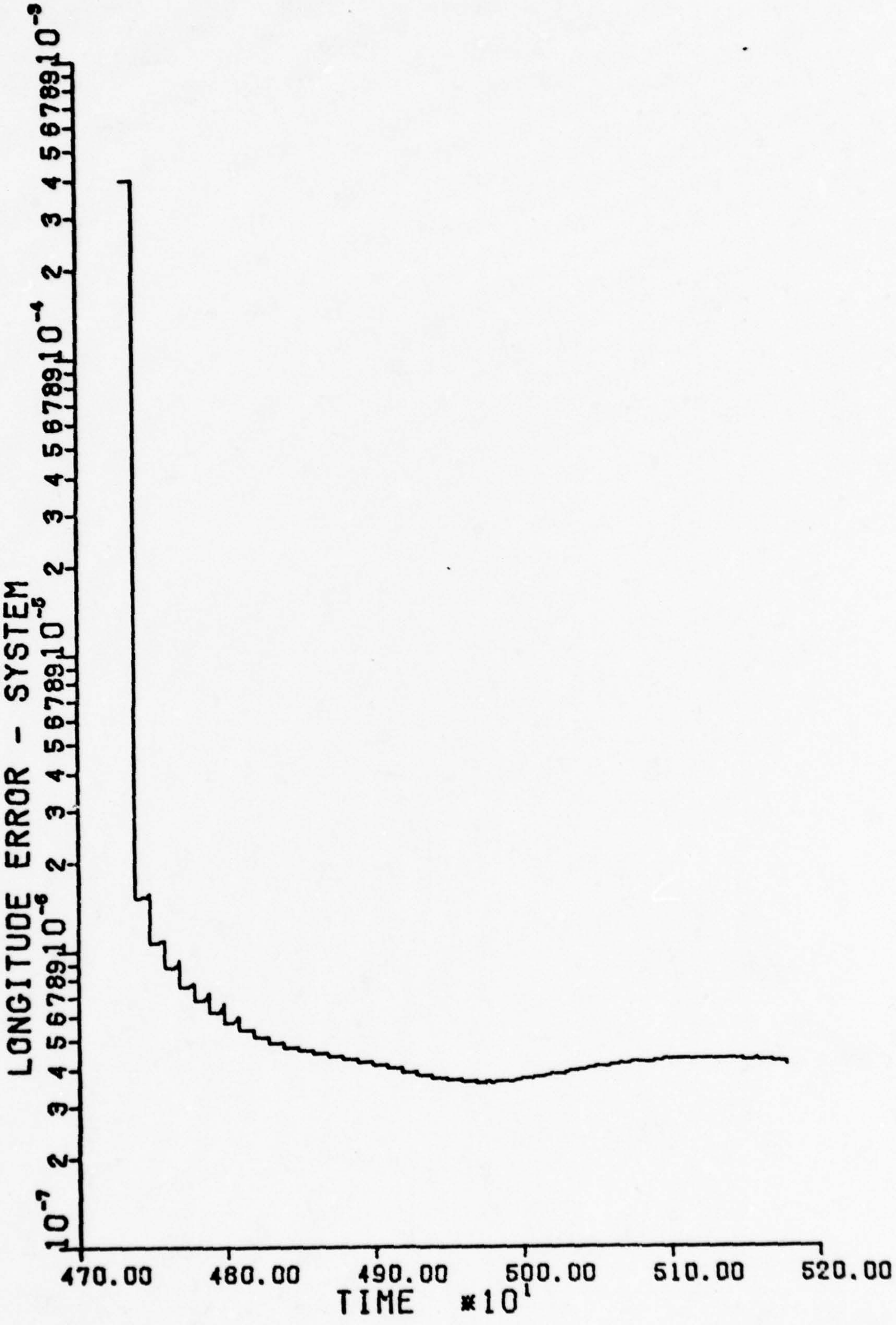


Figure 14. Filter II True Longitude Error

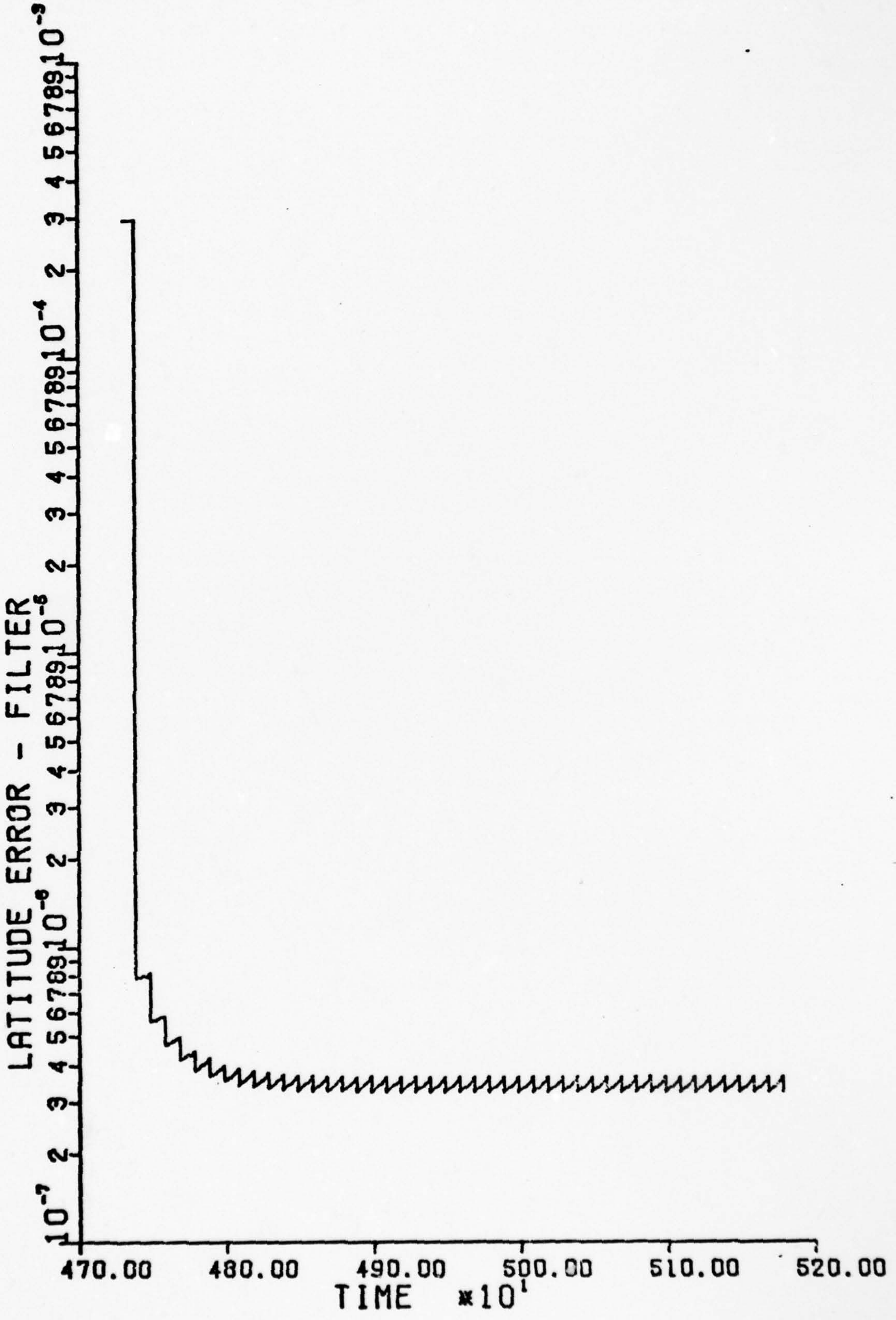


Figure 15. Filter II Estimated Latitude Error

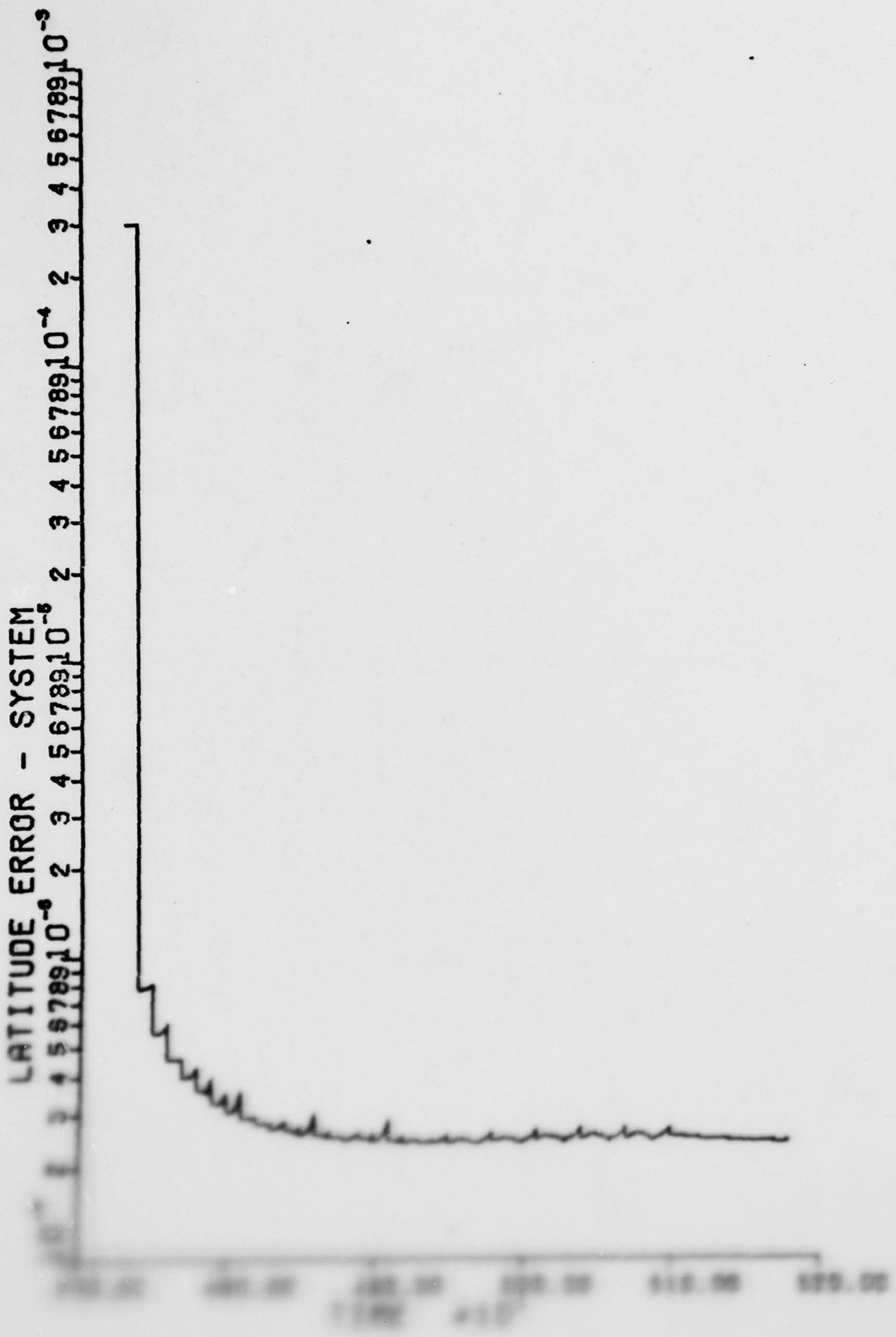


Figure [illegible]. [illegible] Latitude Error

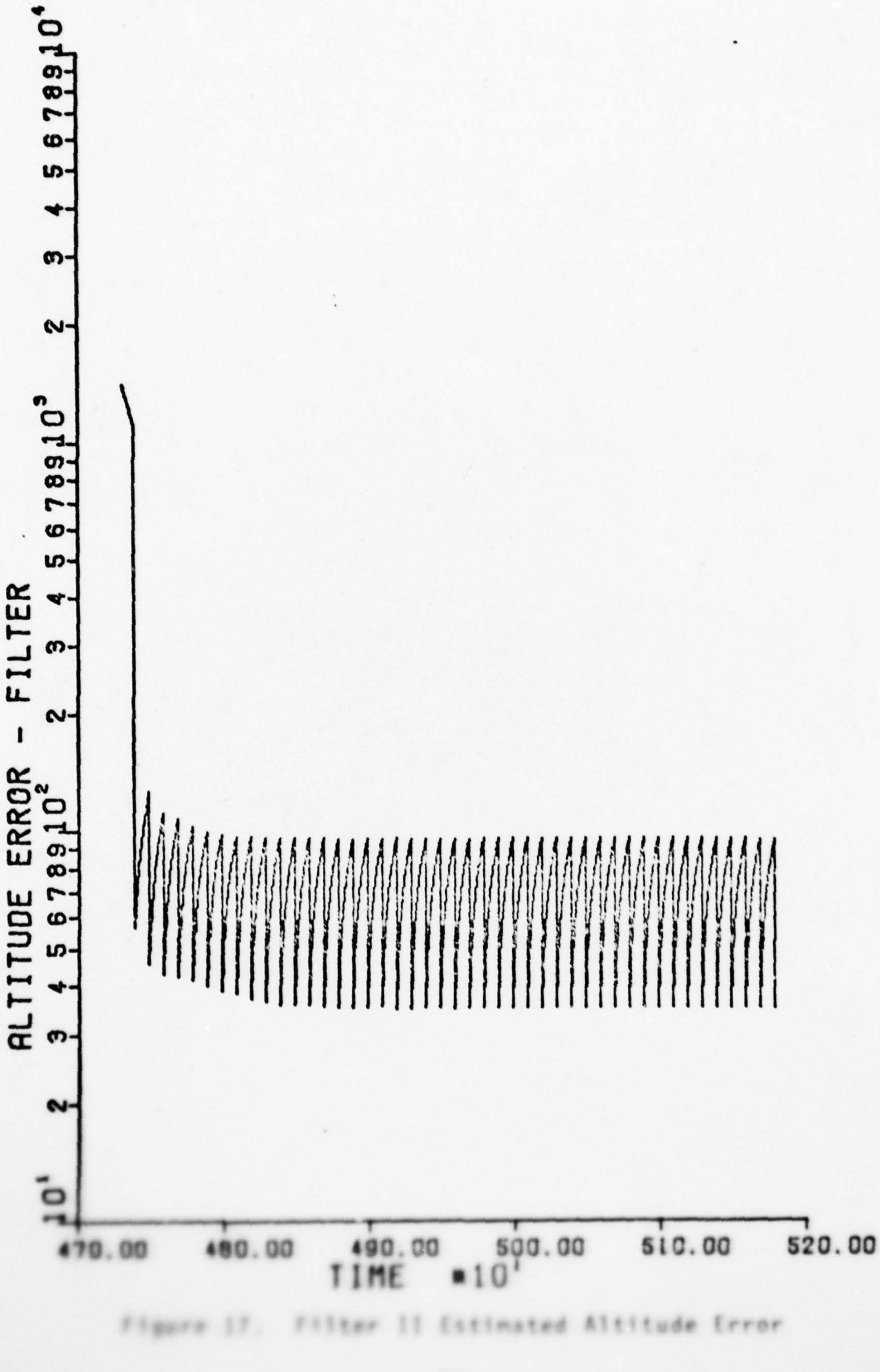


Figure 17. Filter II Estimated Altitude Error

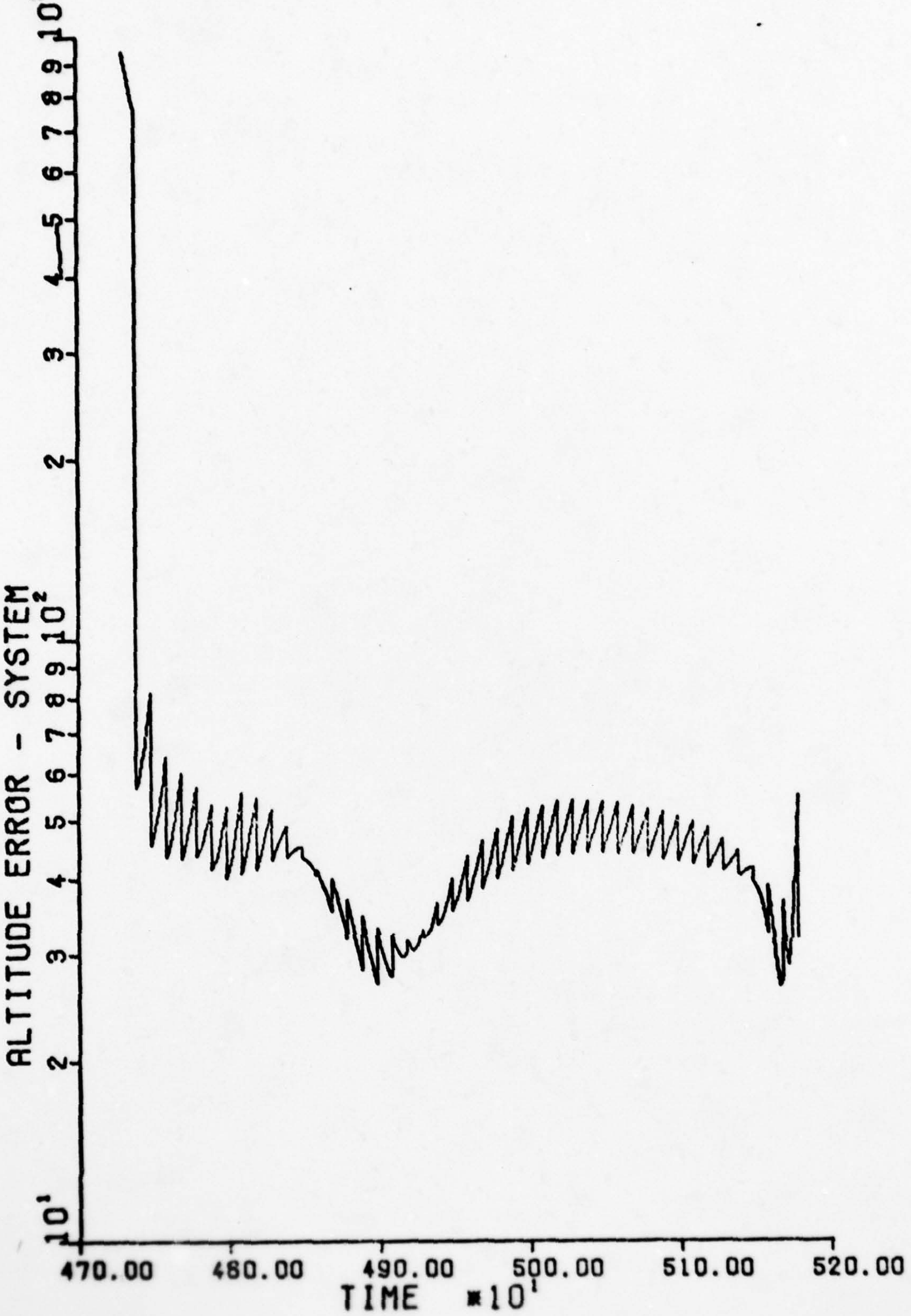


Figure 18. Filter II True Altitude Error

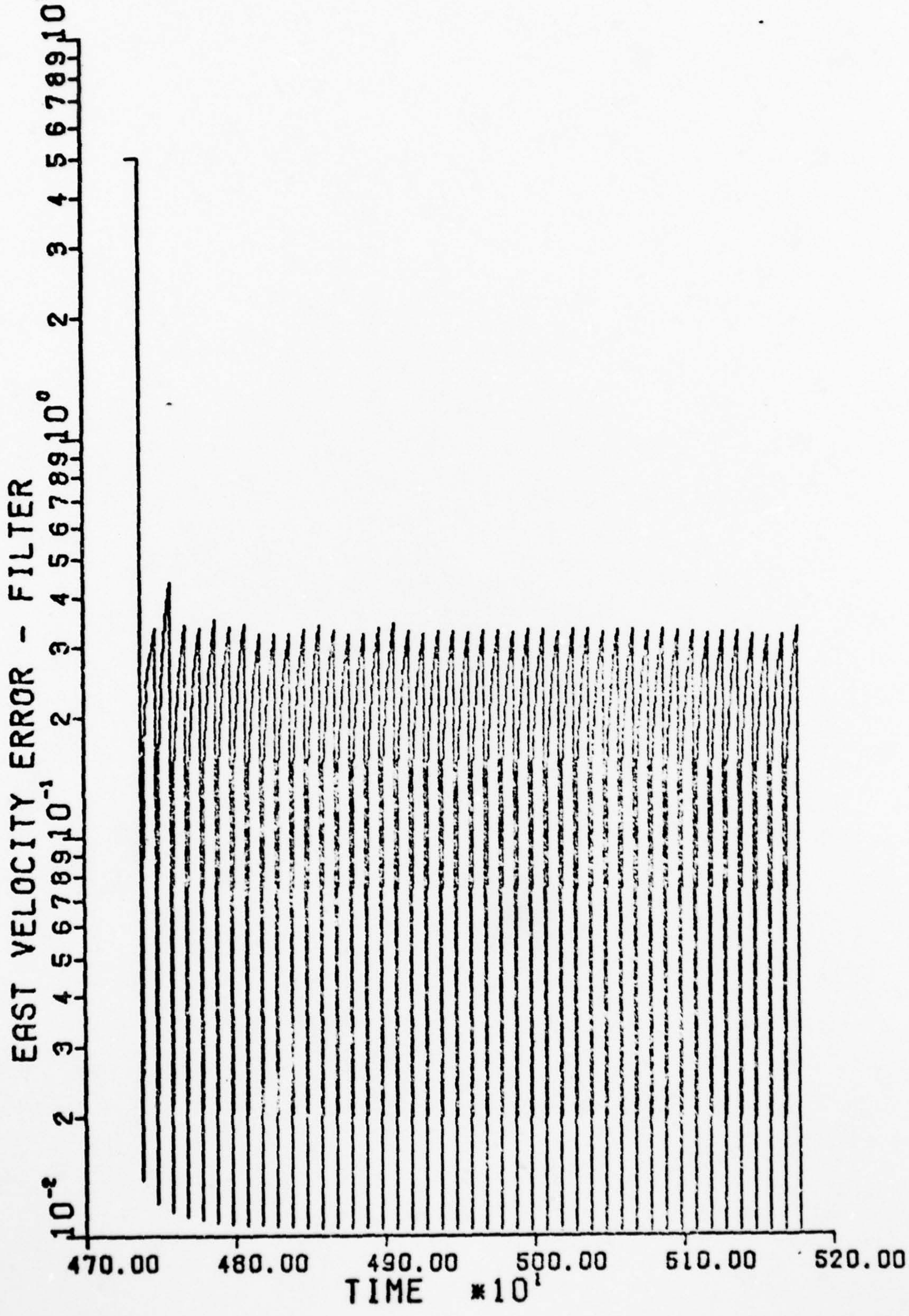


Figure 19. Filter II Estimated East Velocity Error

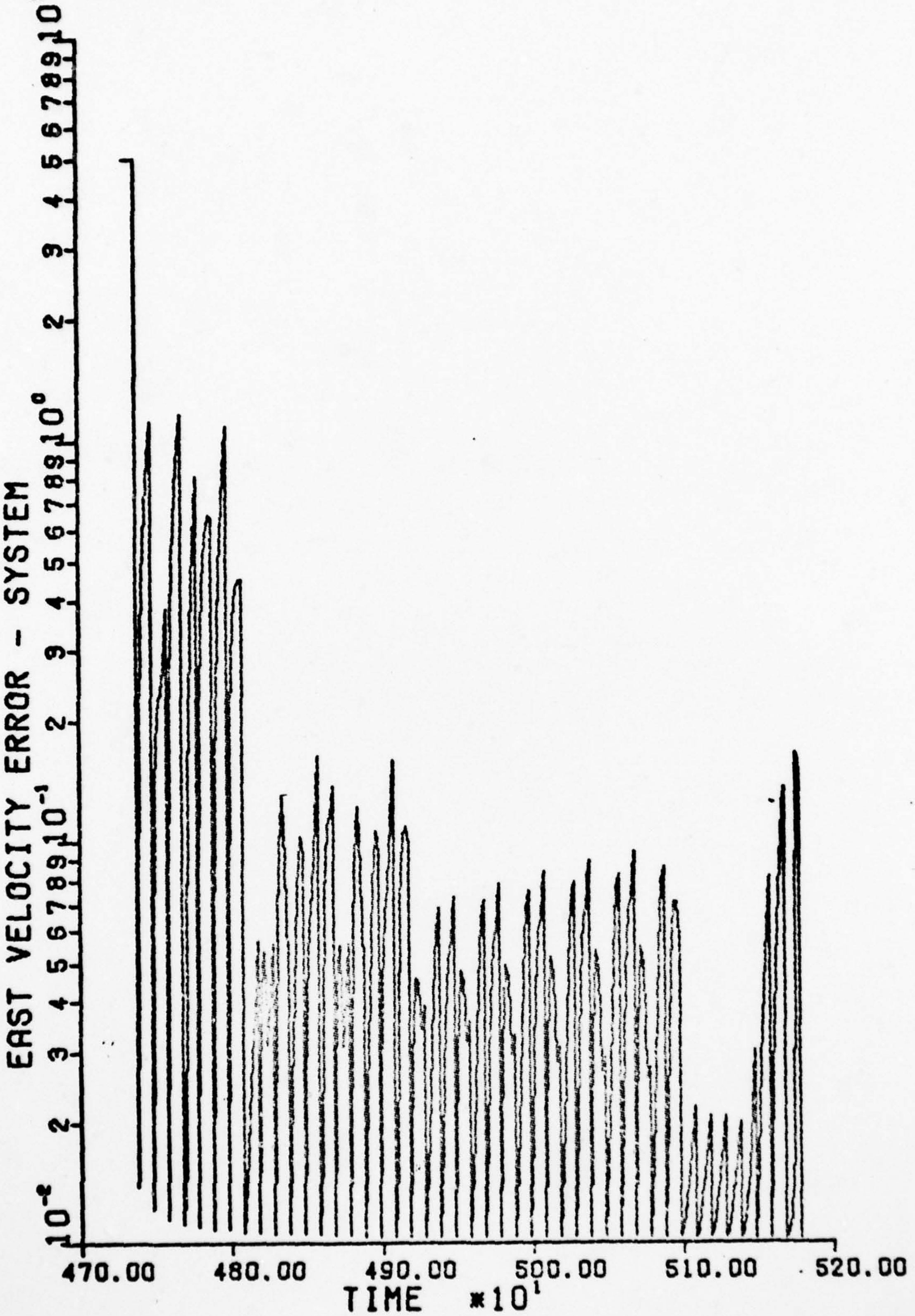


Figure 20. Filter II True East Velocity Error

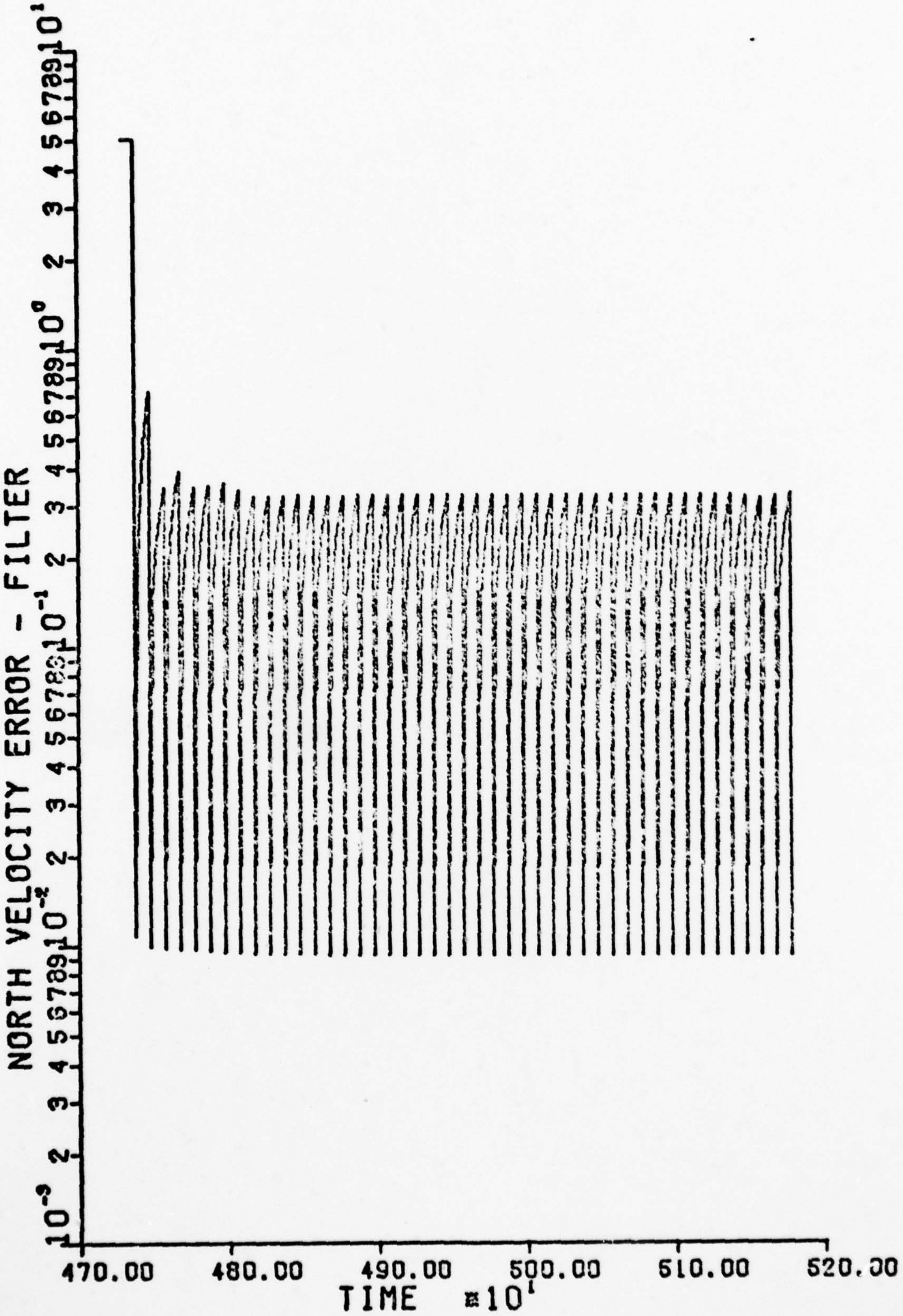


Figure 21. Filter II Estimated North Velocity Error

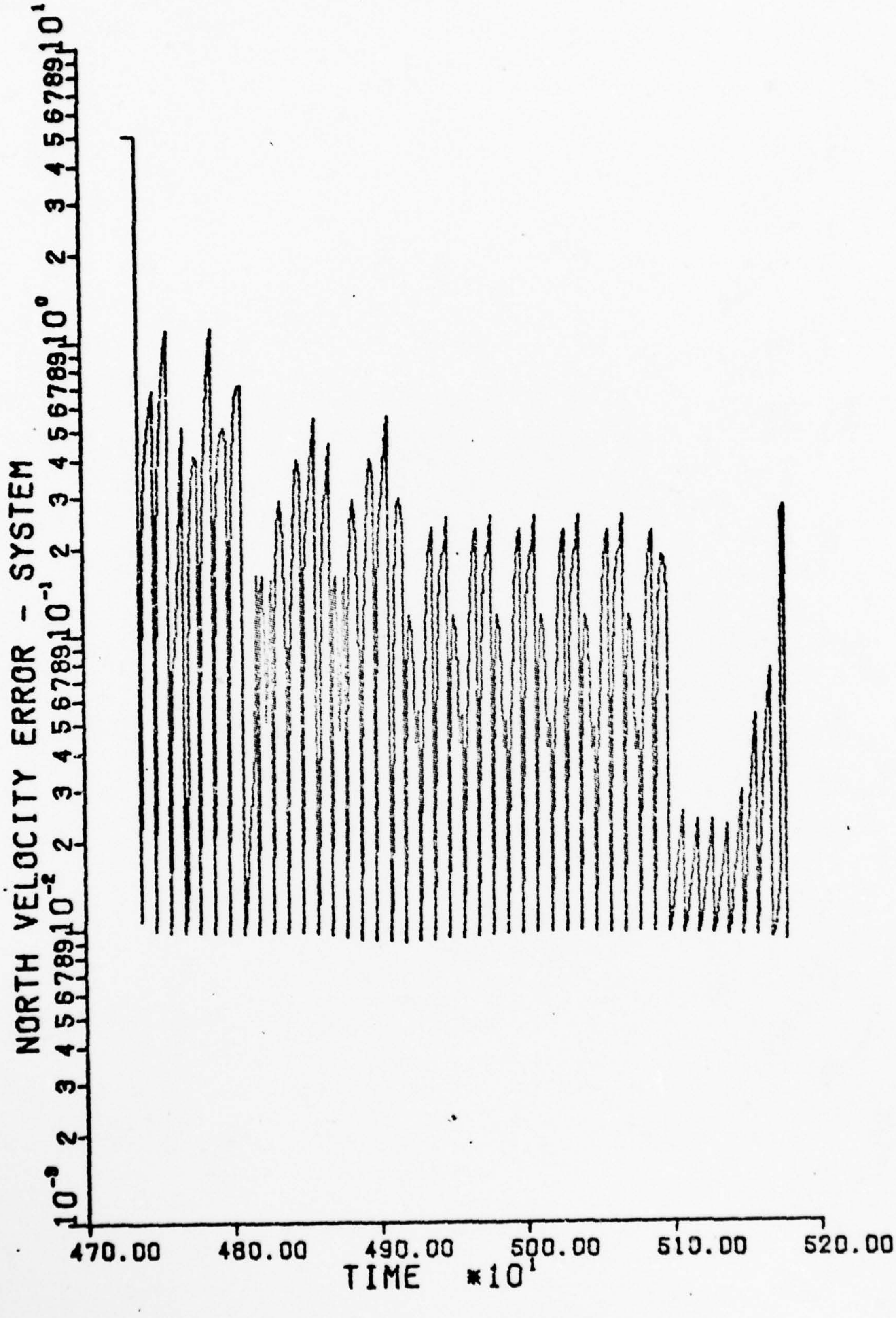


Figure 22. Filter II True North Velocity Error

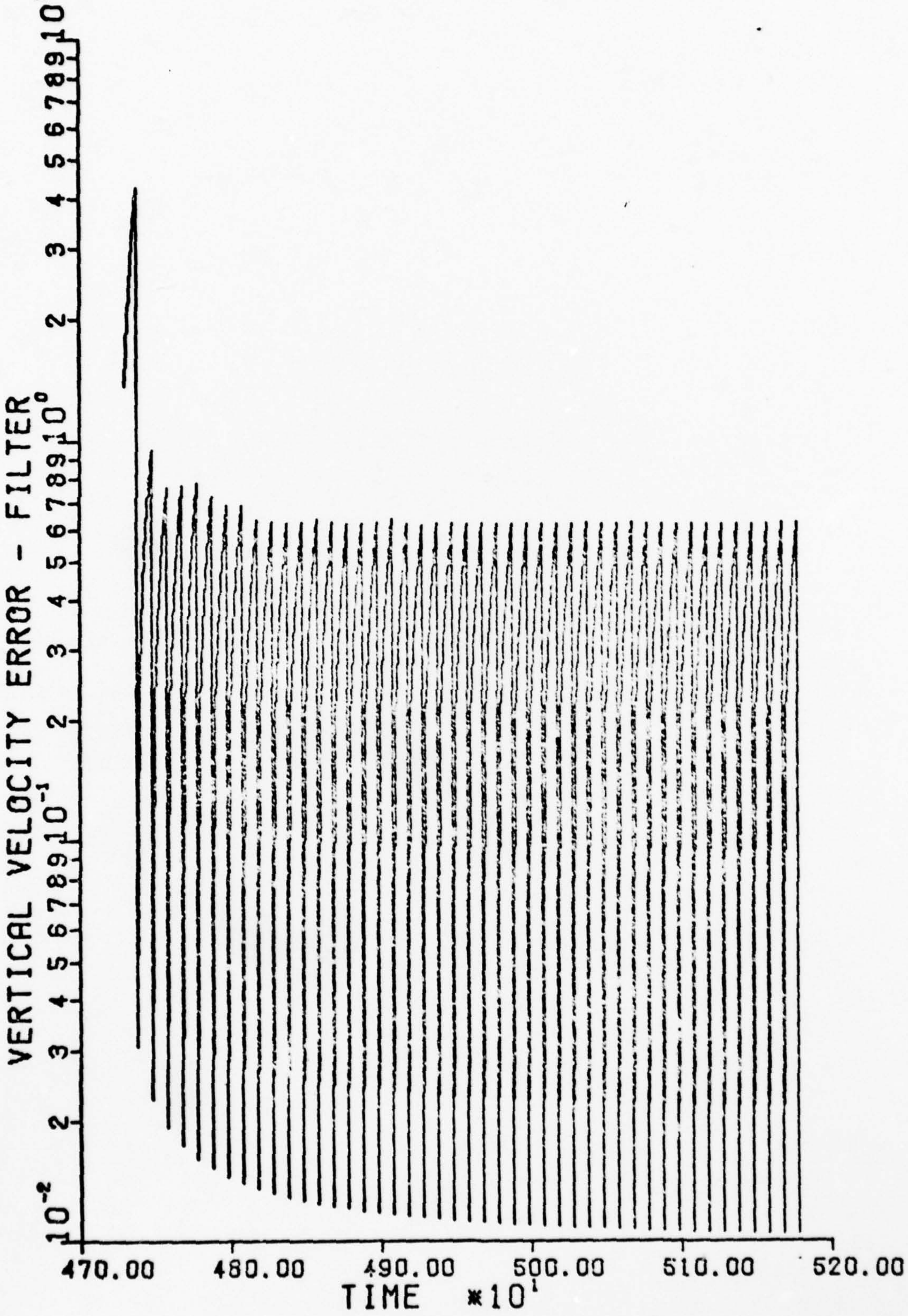


Figure 23. Filter II Estimated Vertical Velocity Error

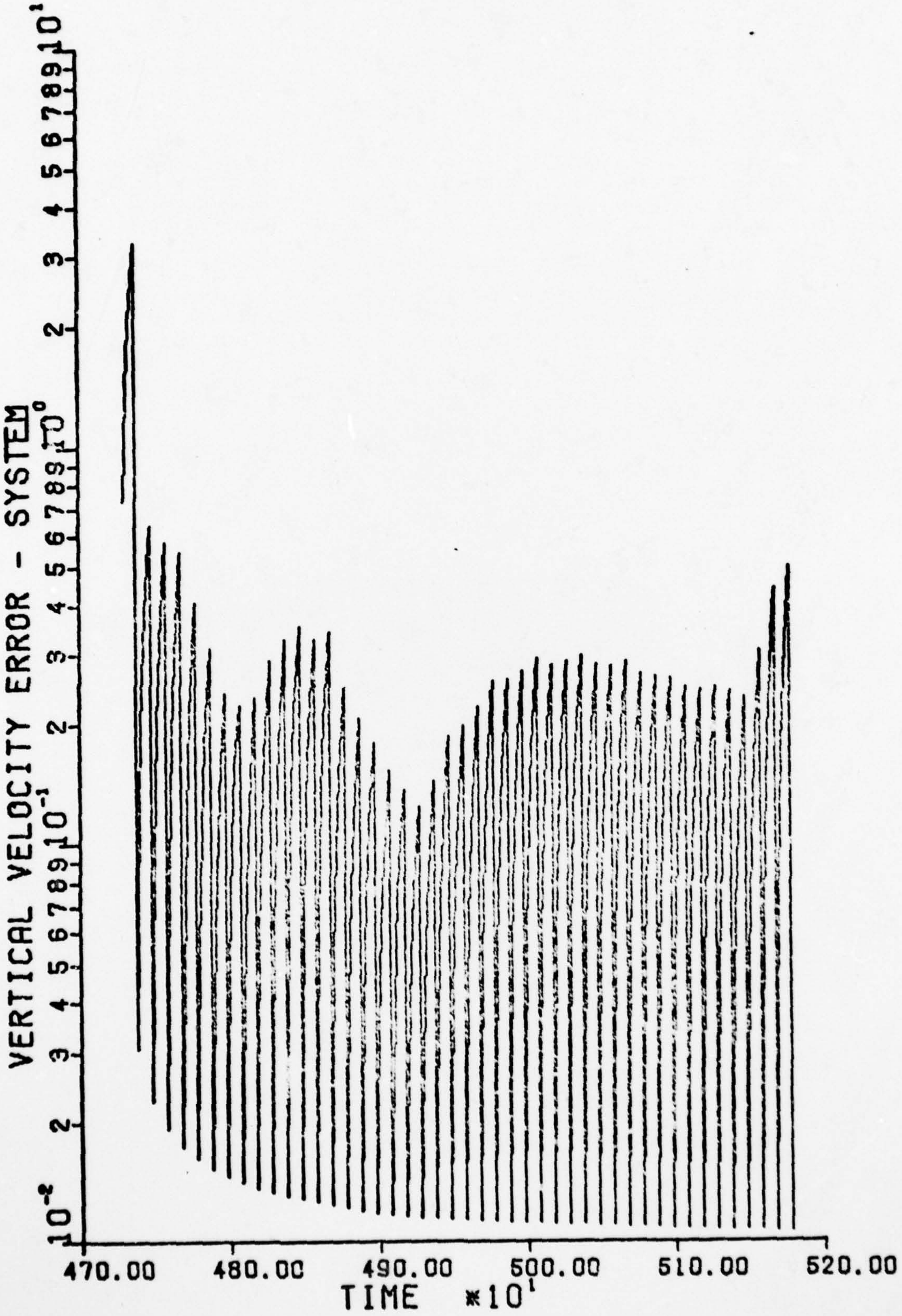


Figure 24. Filter II True Vertical Velocity Error

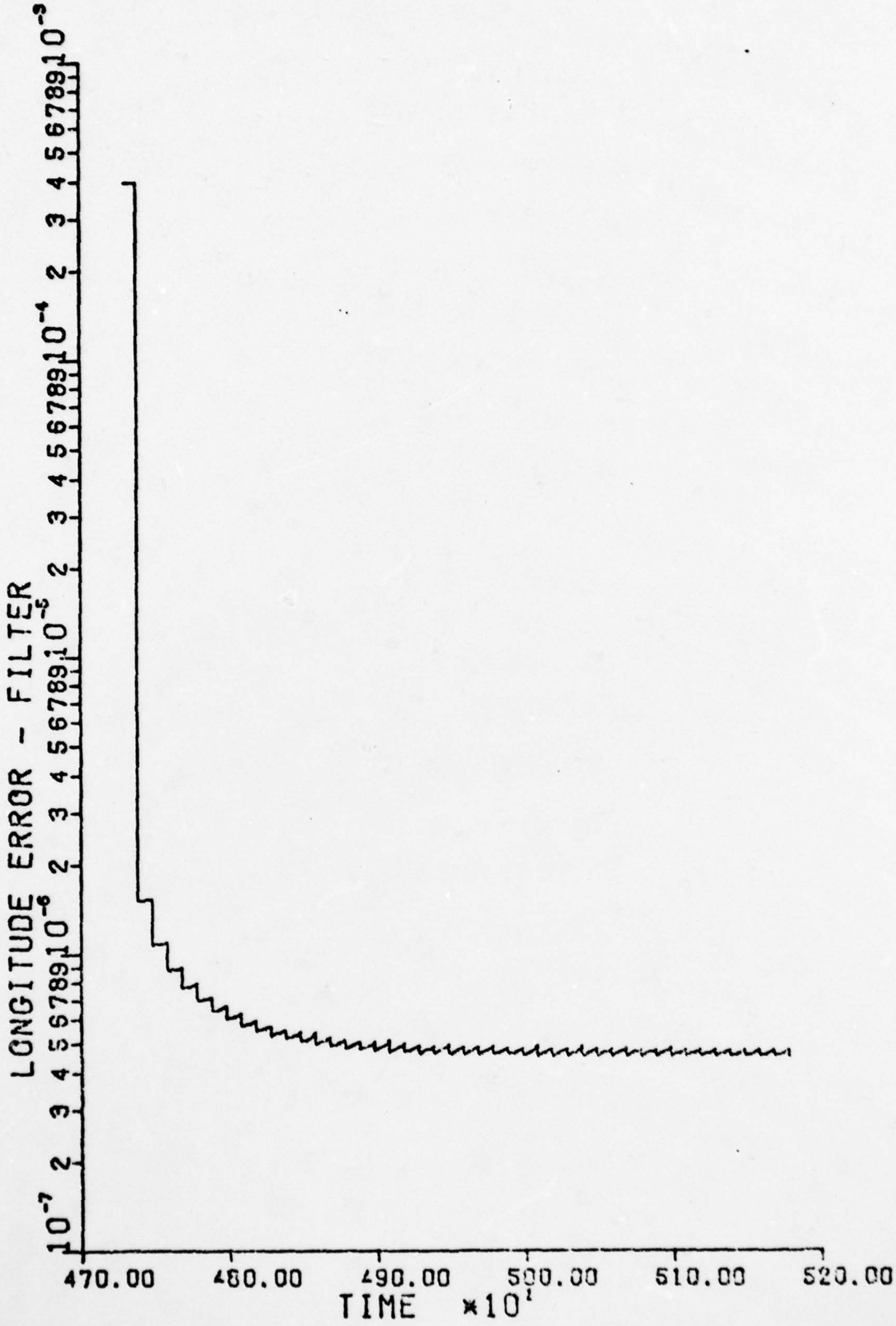


Figure 25. Filter III Estimated Longitude Error

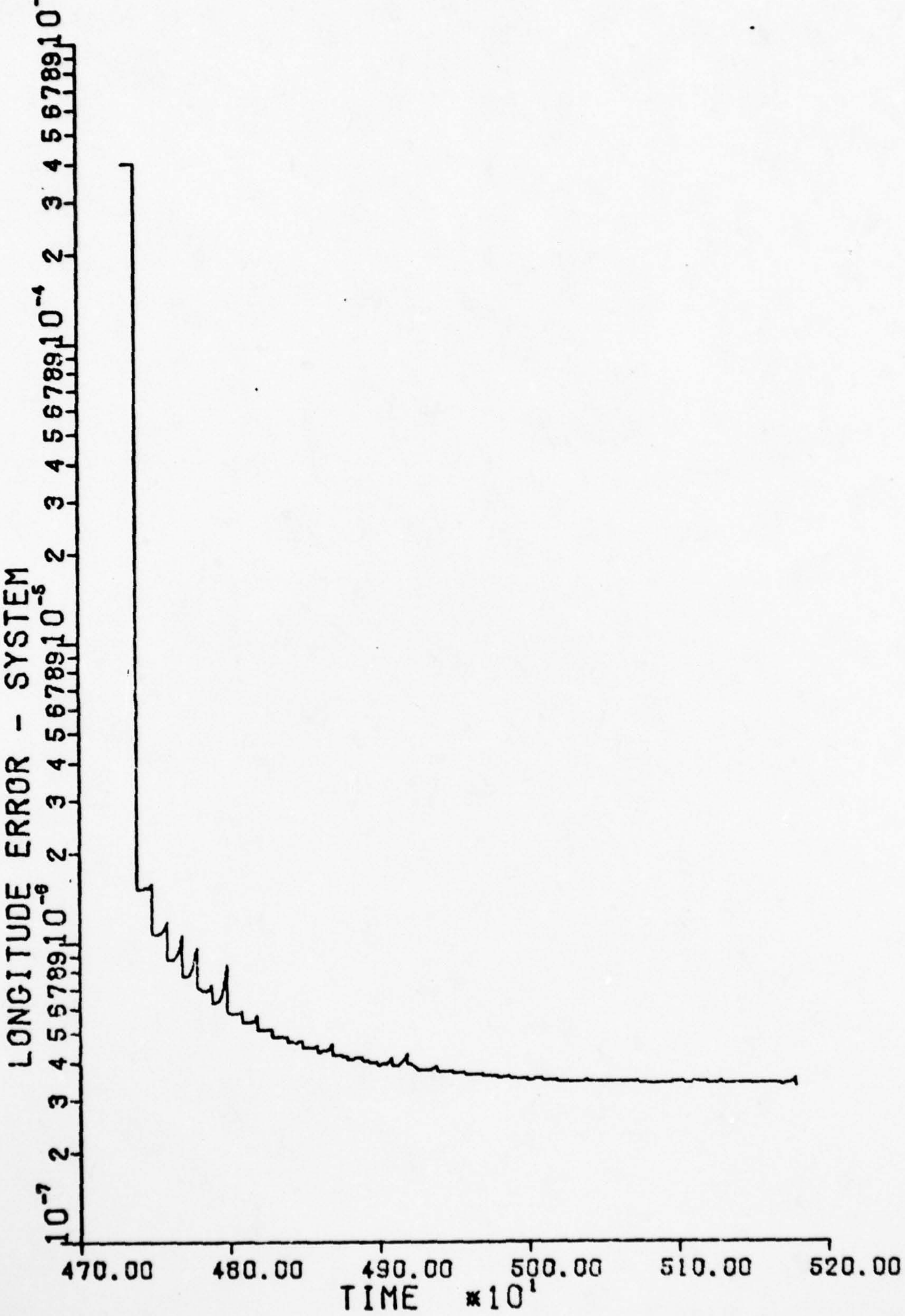


Figure 26. Filter III True Longitude Error

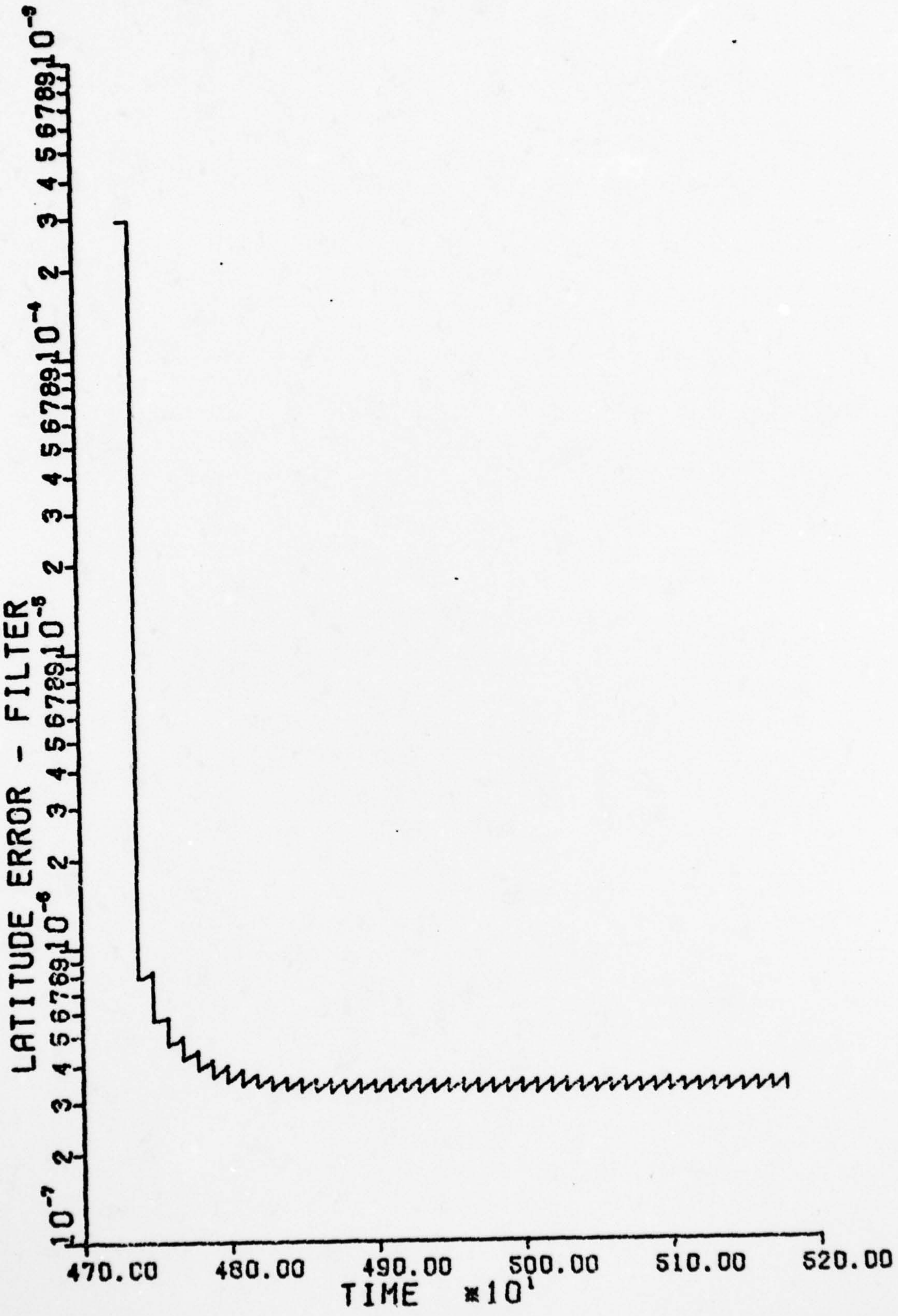


Figure 27. Filter III Estimated Latitude Error

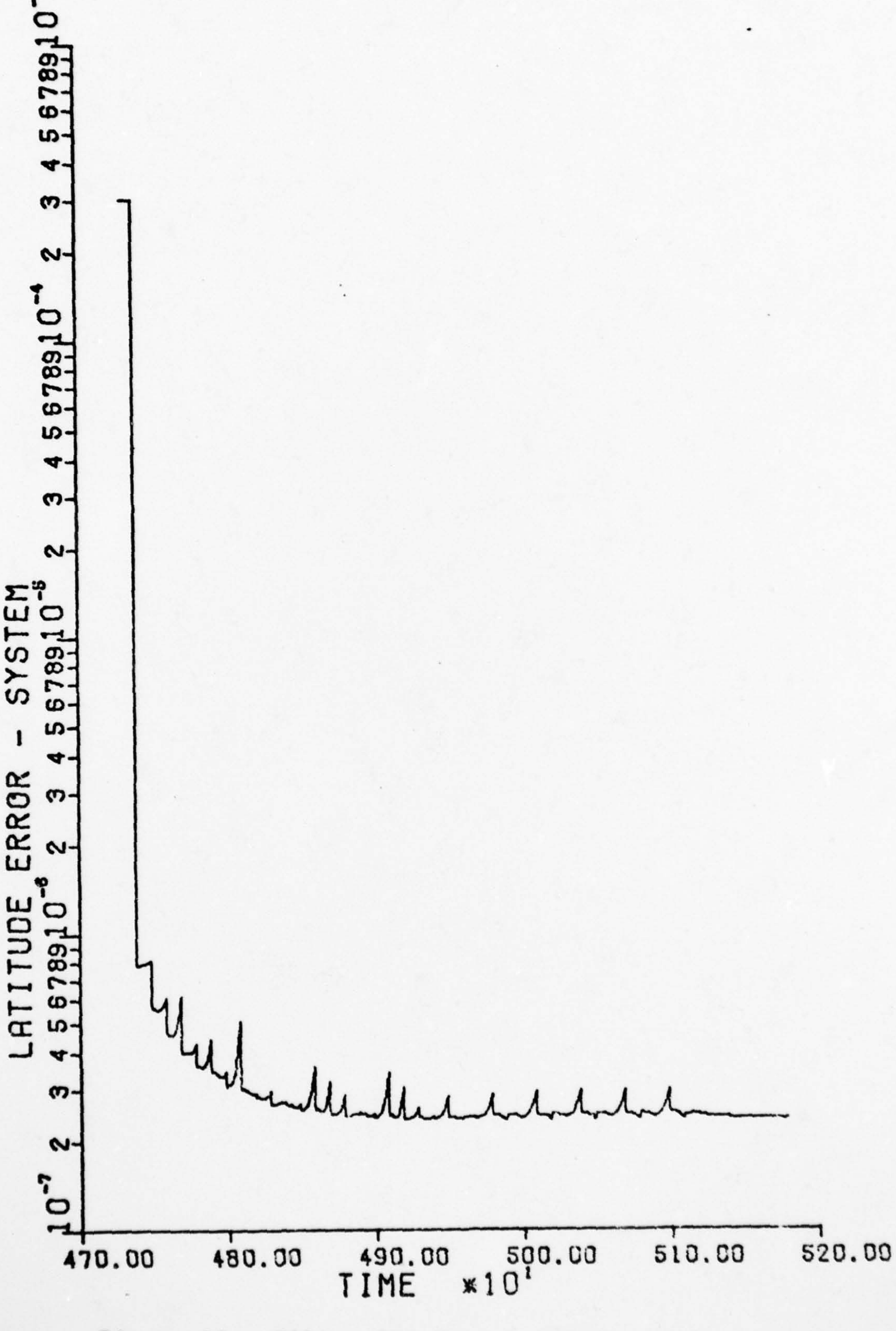


Figure 28. Filter III True Latitude Error

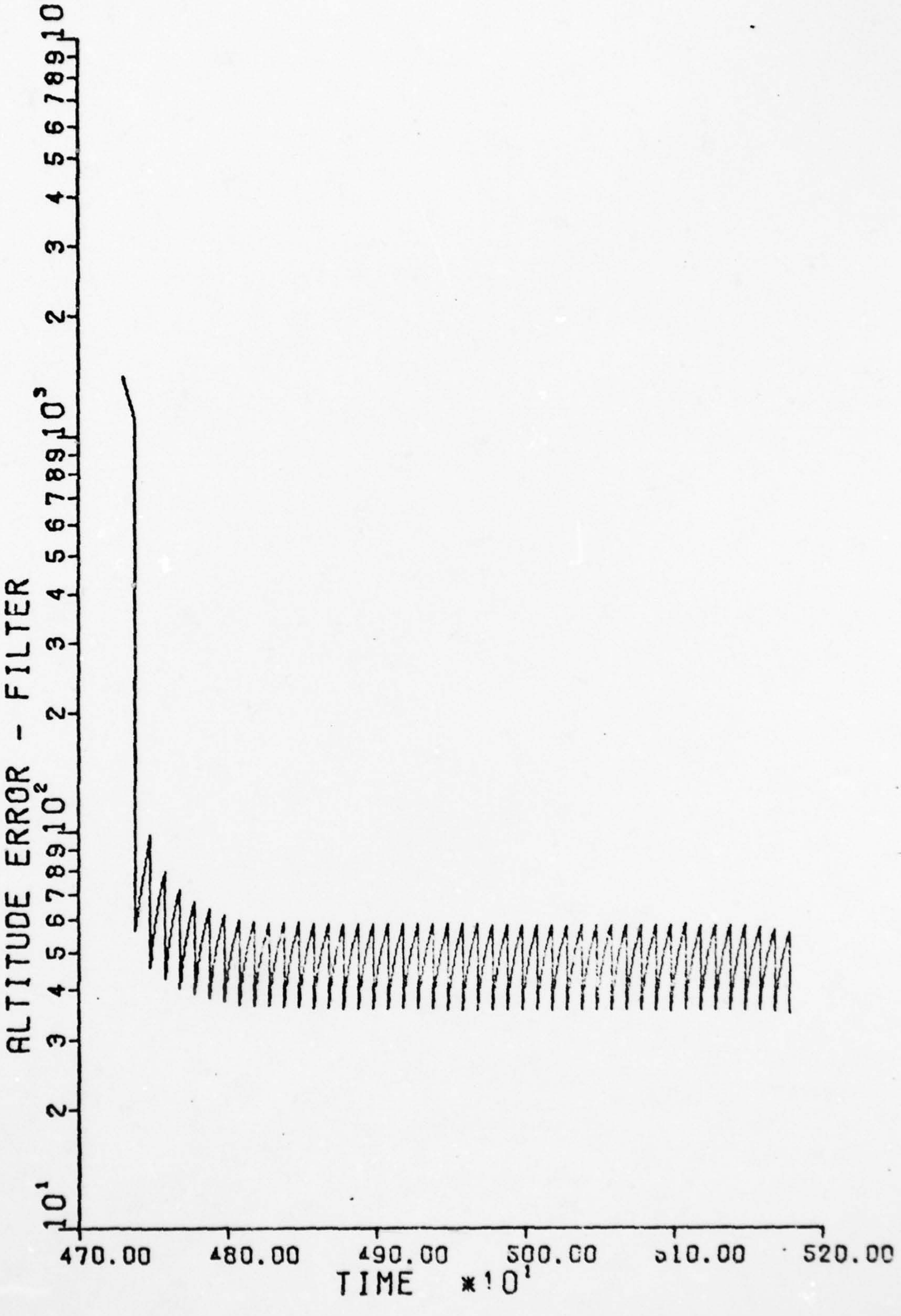
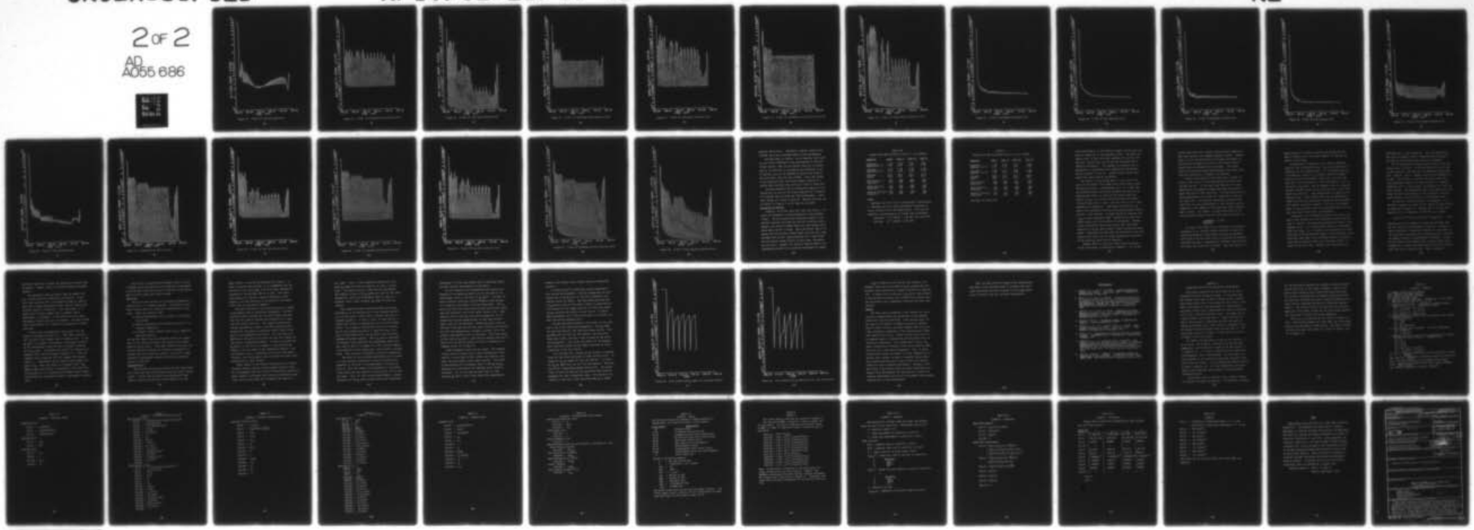


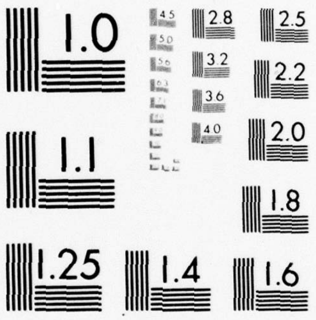
Figure 29. Filter III Estimated Altitude Error

AD-A055 686 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 17/7
STATE NOISE COVARIANCE COMPUTATION IN THE KALMAN FILTER.(U)
DEC 77 D A ARPIN
UNCLASSIFIED AFIT/GE/EE/77-42 NL

2 OF 2
AD
A055 686



END
DATE
FILMED
8-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

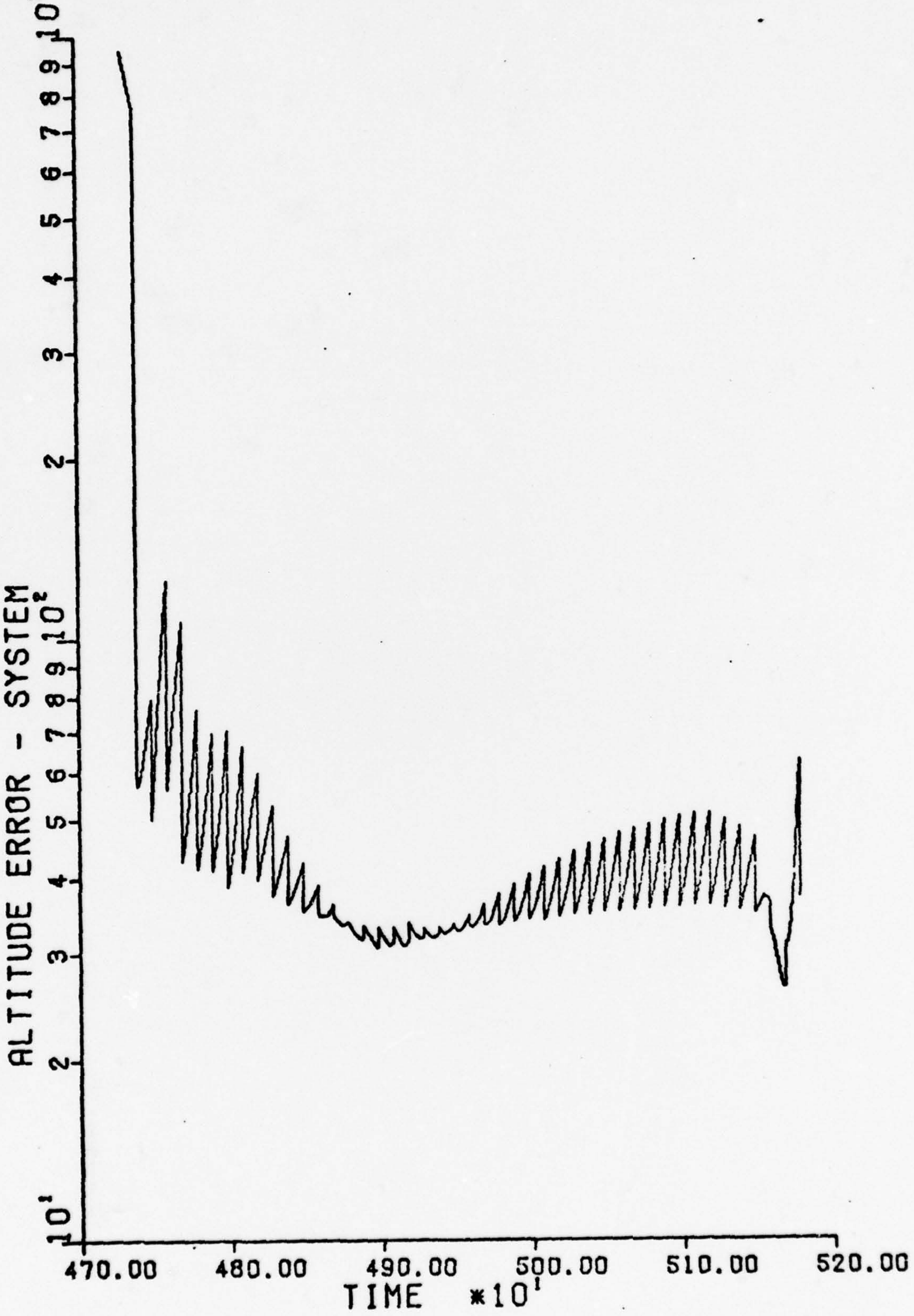


Figure 30. Filter III True Altitude Error

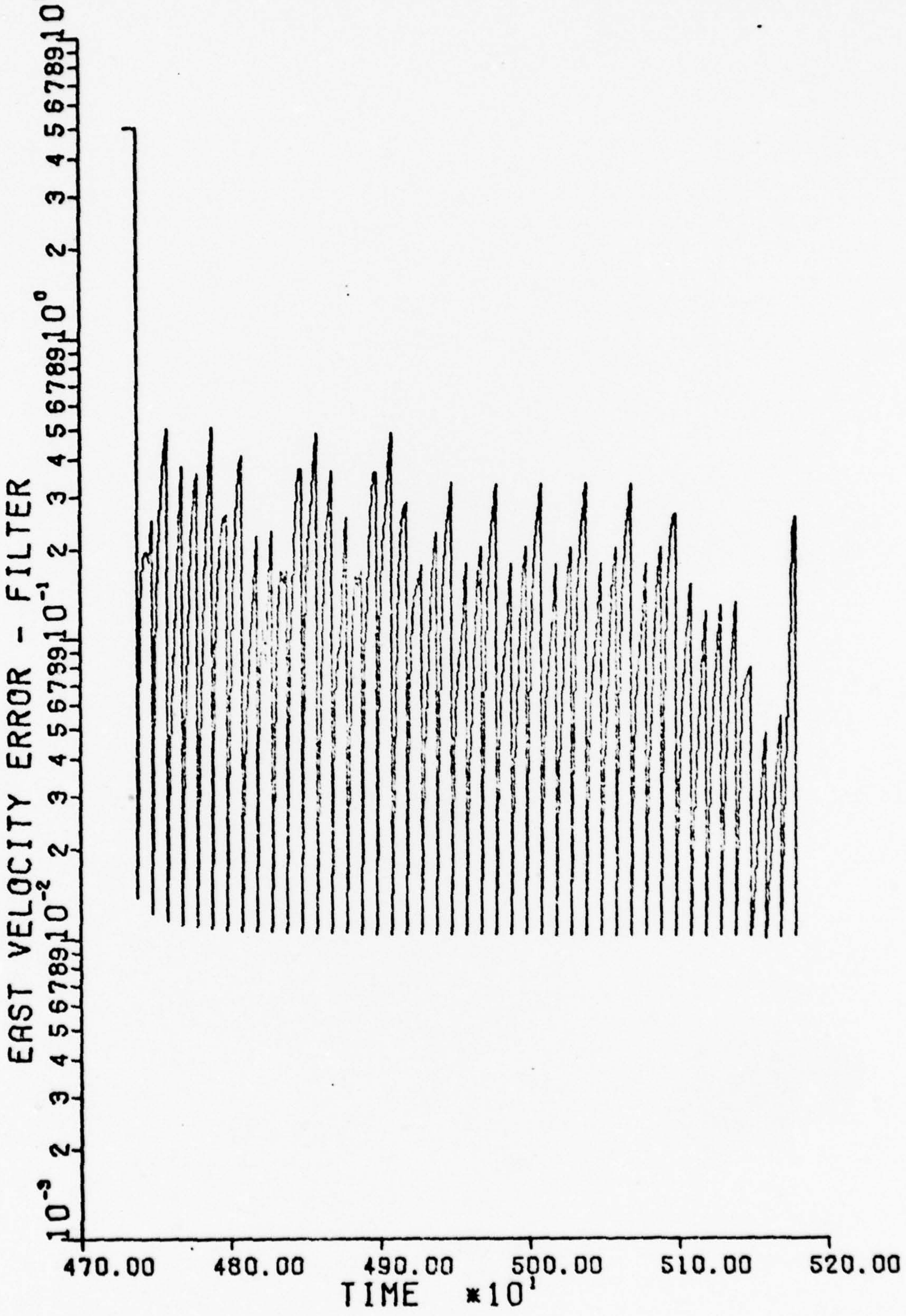


Figure 31. Filter III Estimated East Velocity Error

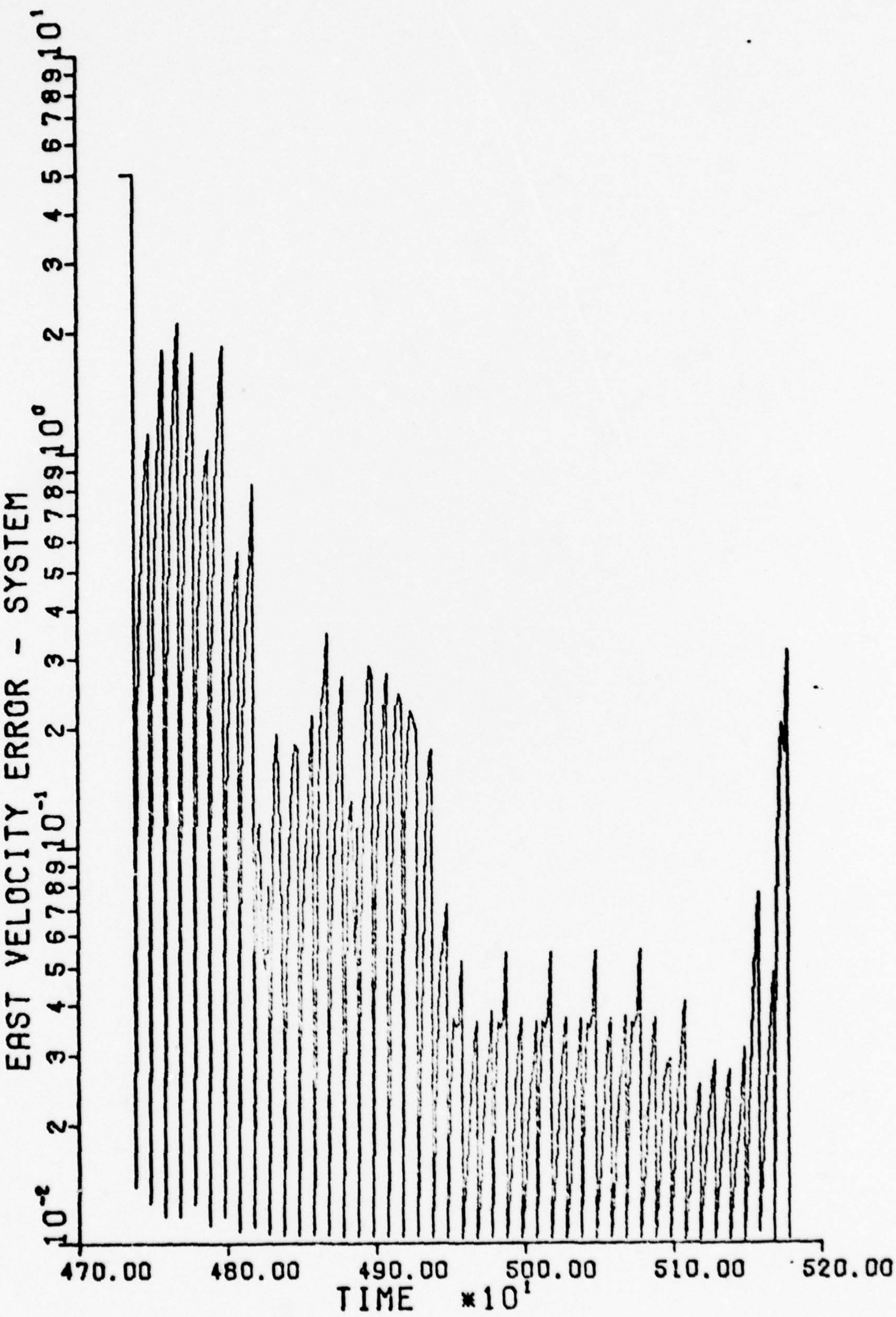


Figure 32. Filter III True East Velocity Error

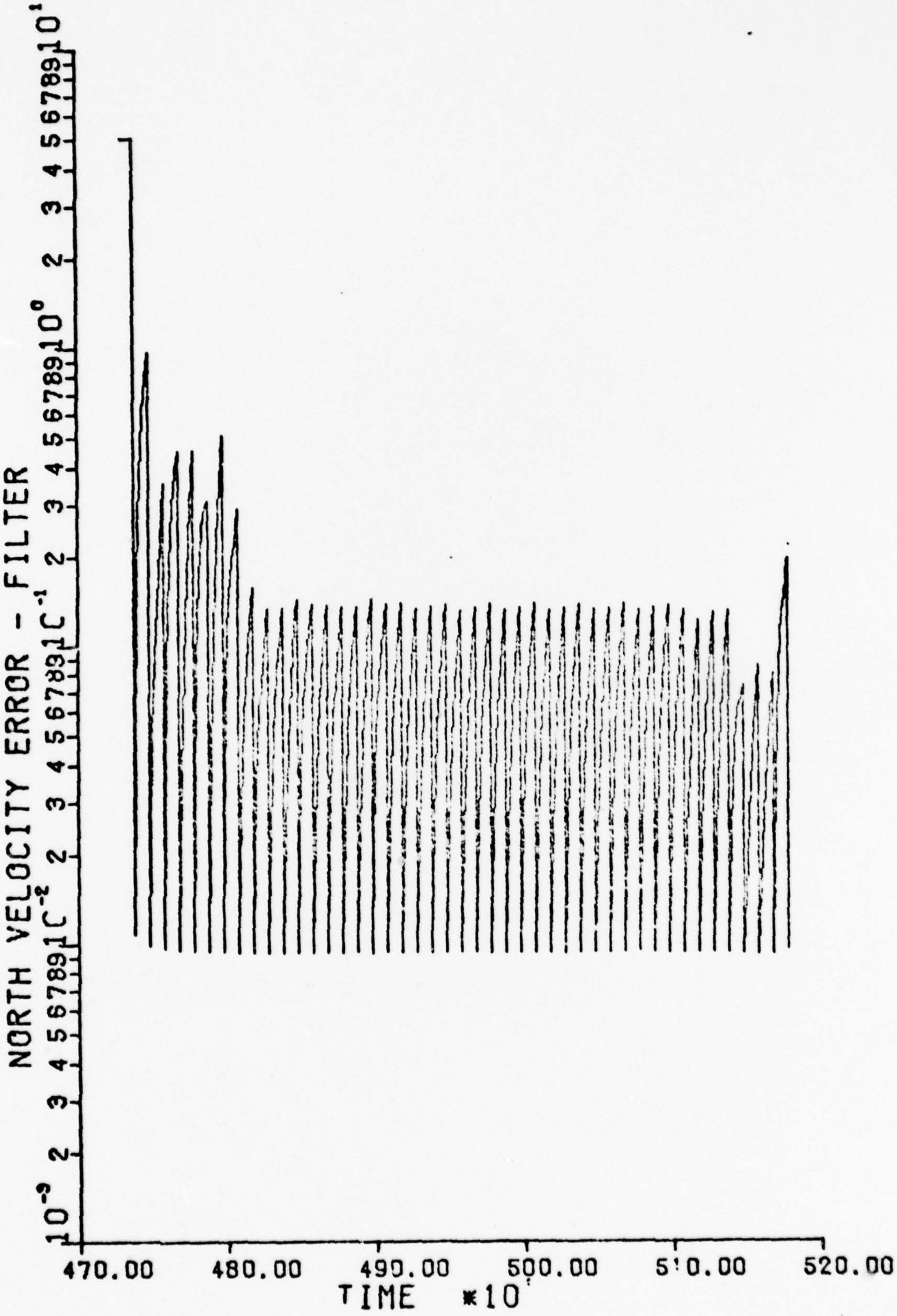


Figure 33. Filter III Estimated North Velocity Error

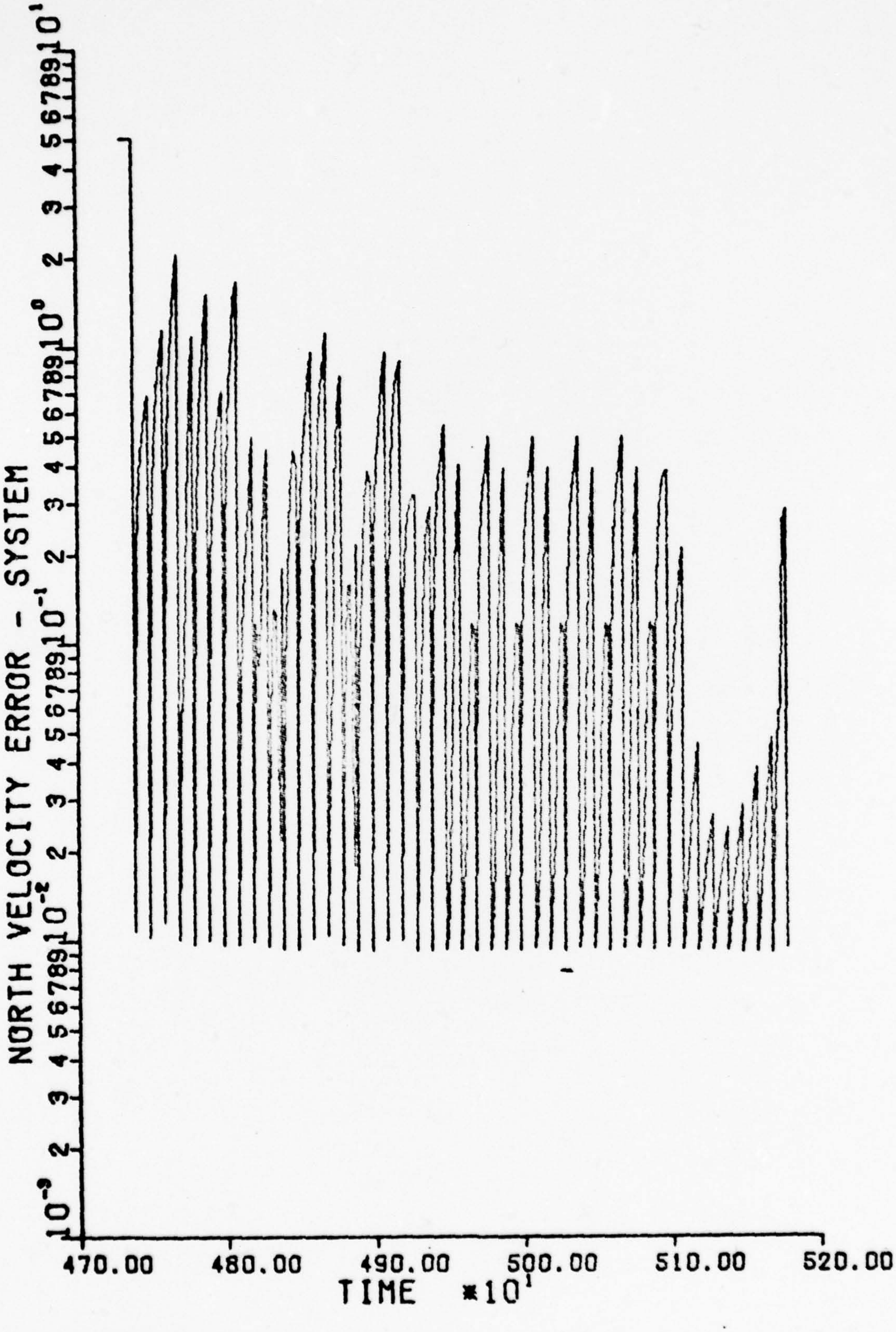


Figure 34. Filter III True North Velocity Error

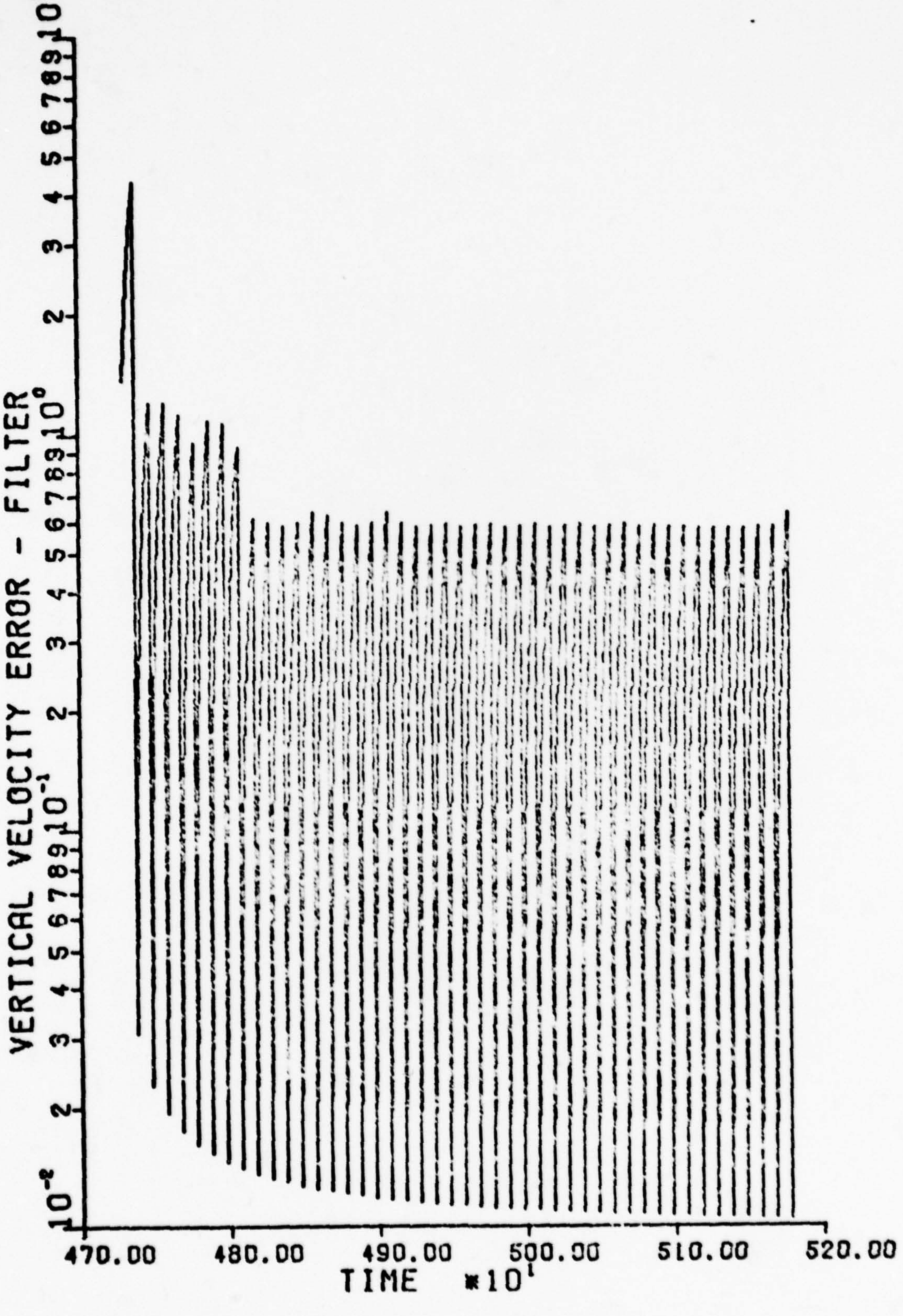


Figure 35. Filter III Estimated Vertical Velocity Error

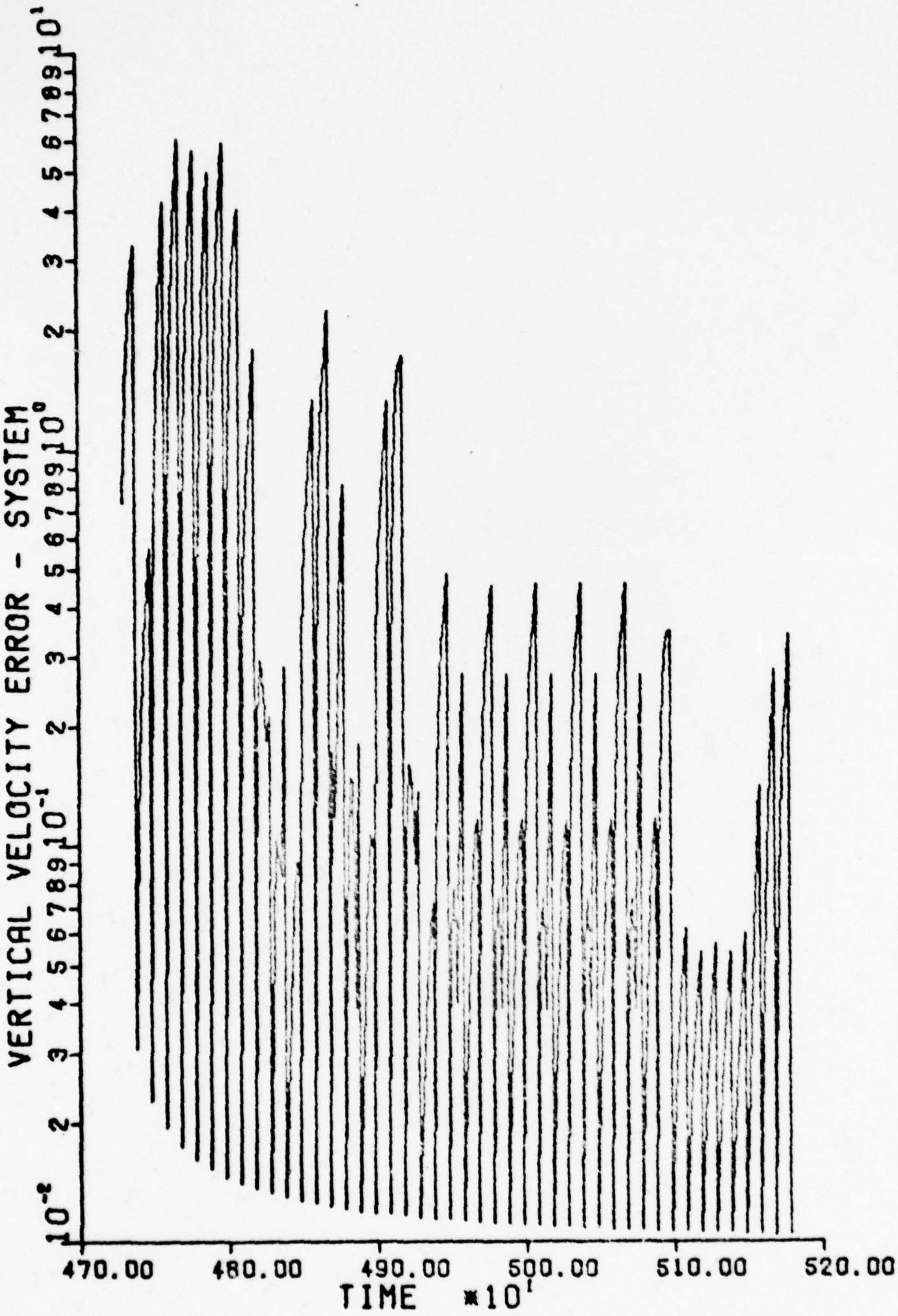


Figure 36. Filter III True Vertical Velocity Error

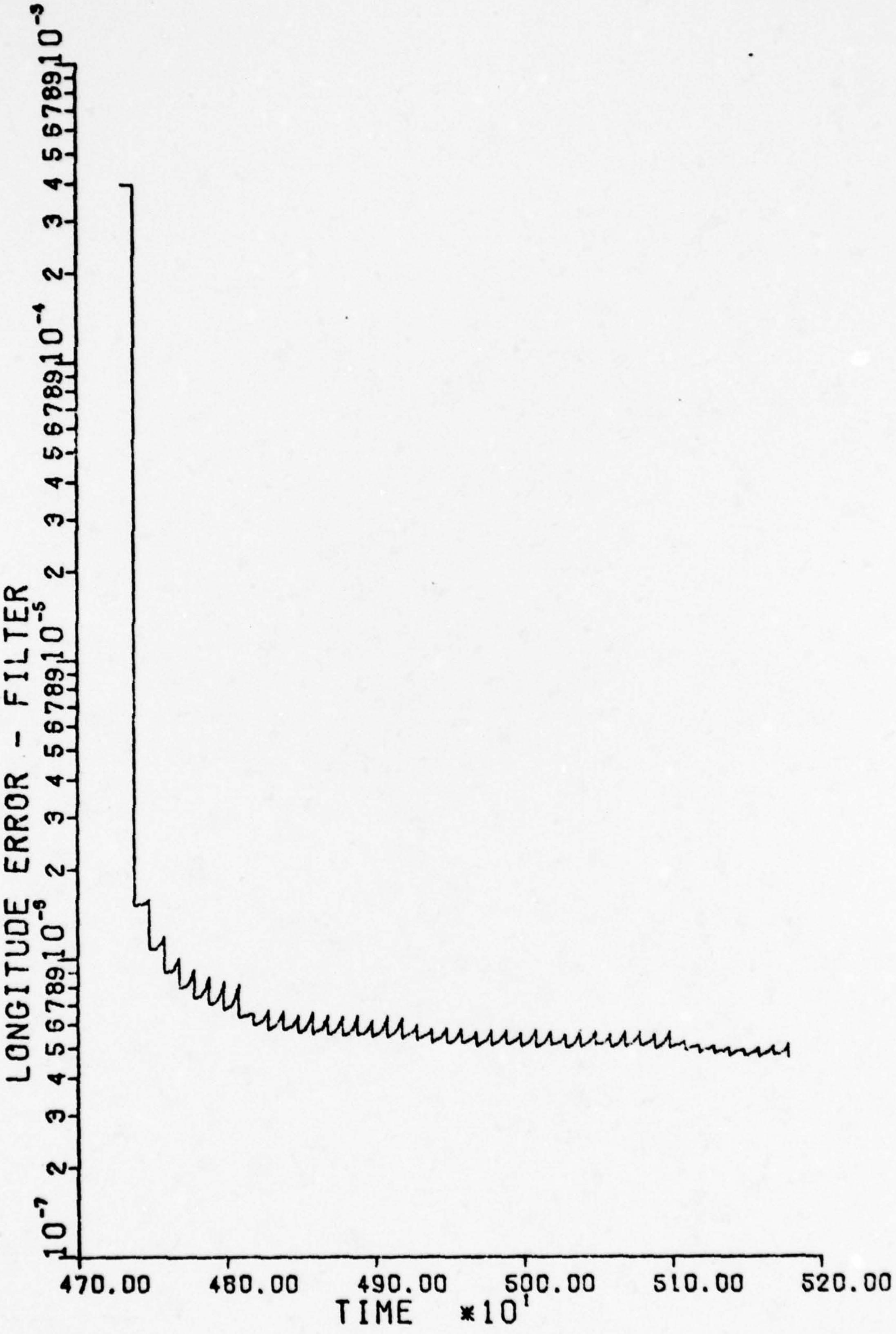


Figure 37. Filter IV Estimated Longitude Error

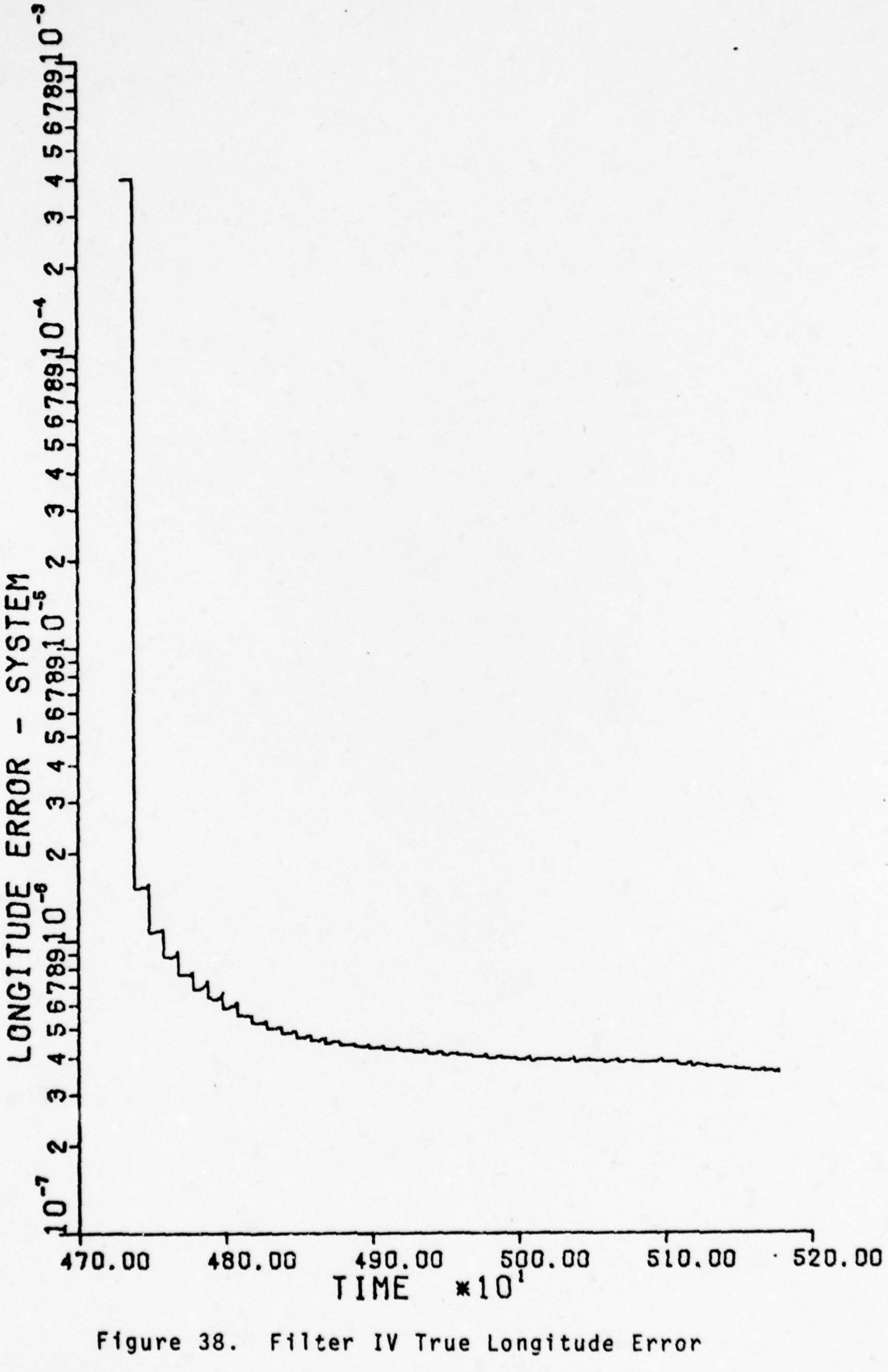


Figure 38. Filter IV True Longitude Error

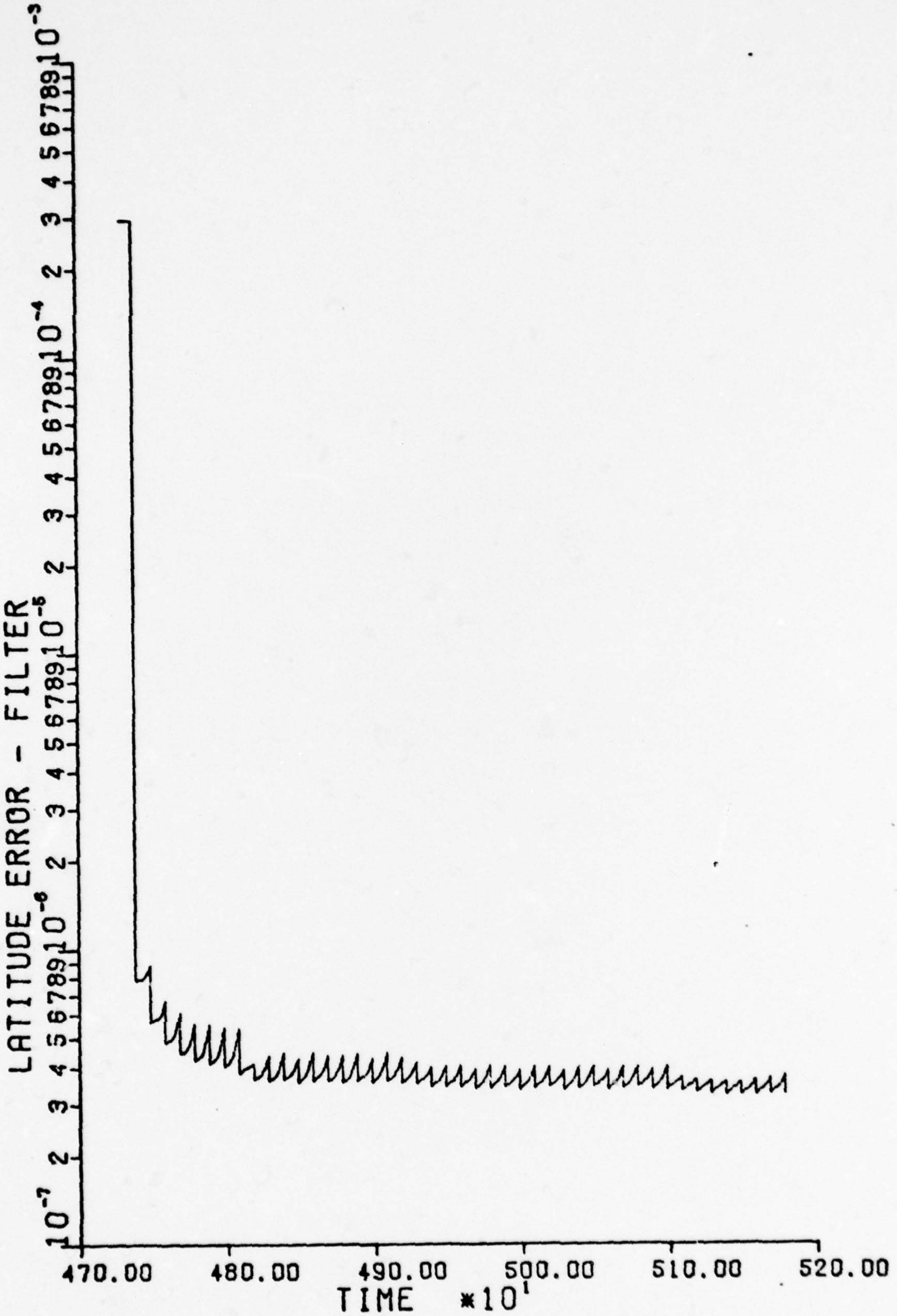


Figure 39. Filter IV Estimated Latitude Error

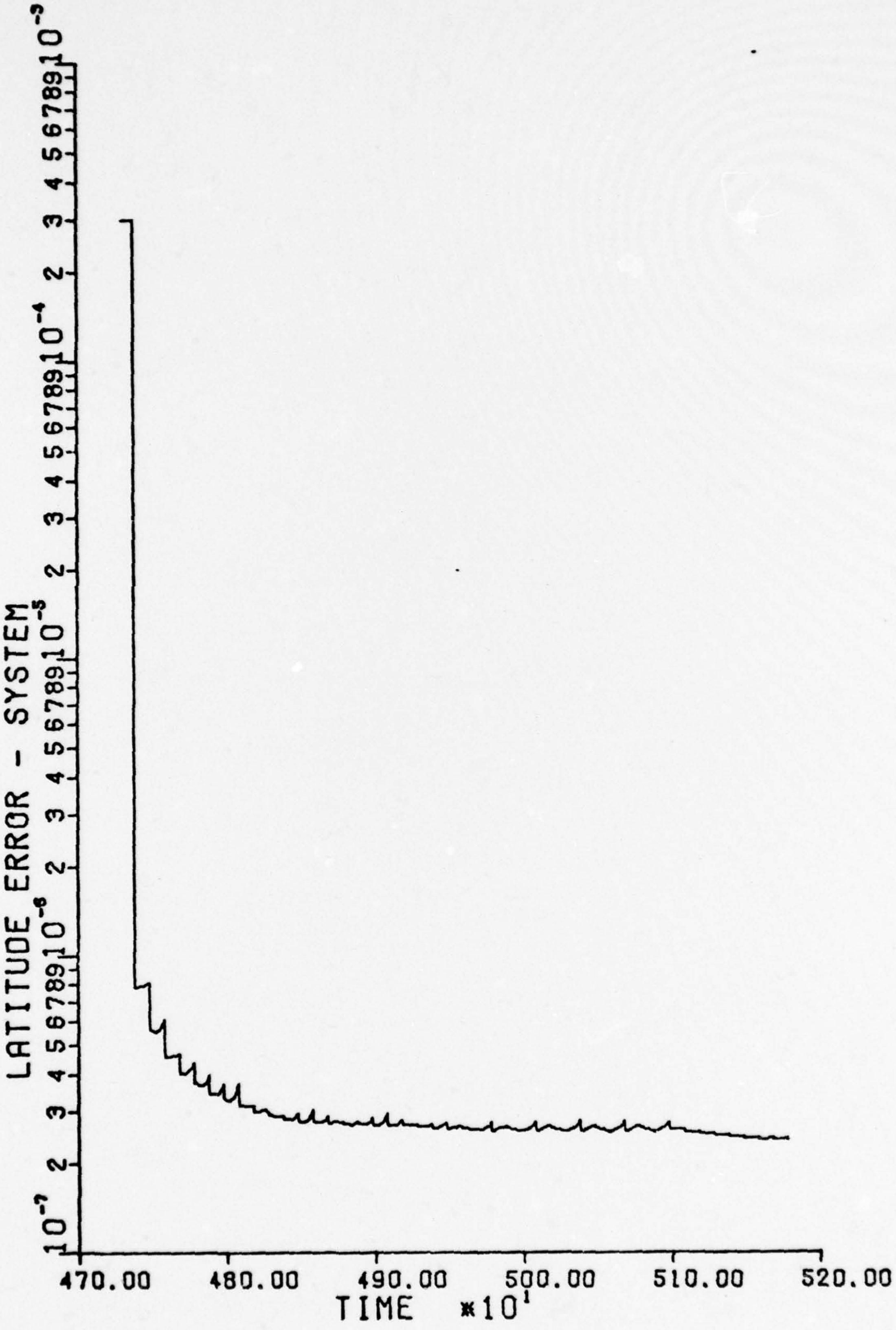


Figure 40. Filter IV True Latitude Error

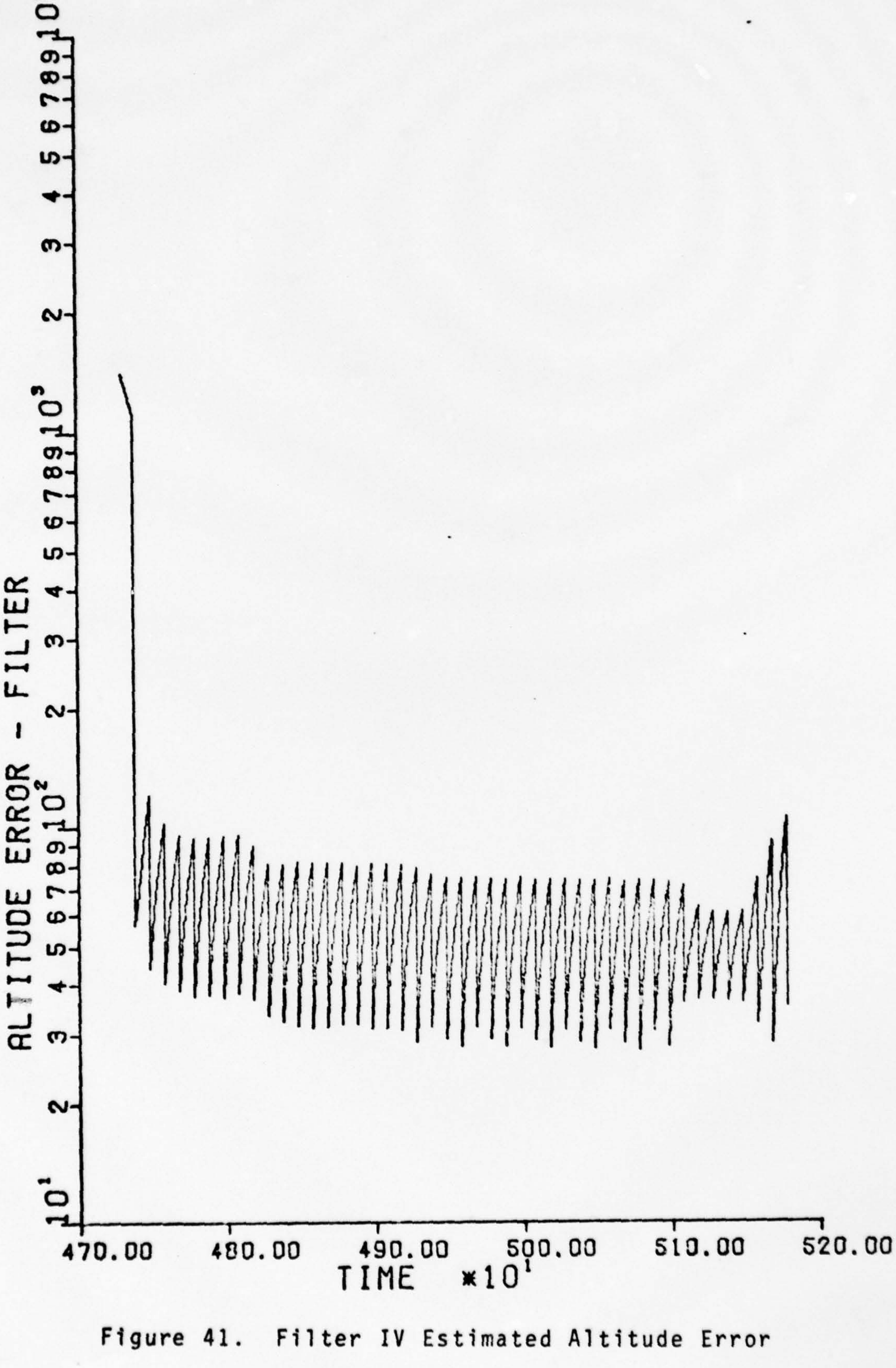


Figure 41. Filter IV Estimated Altitude Error

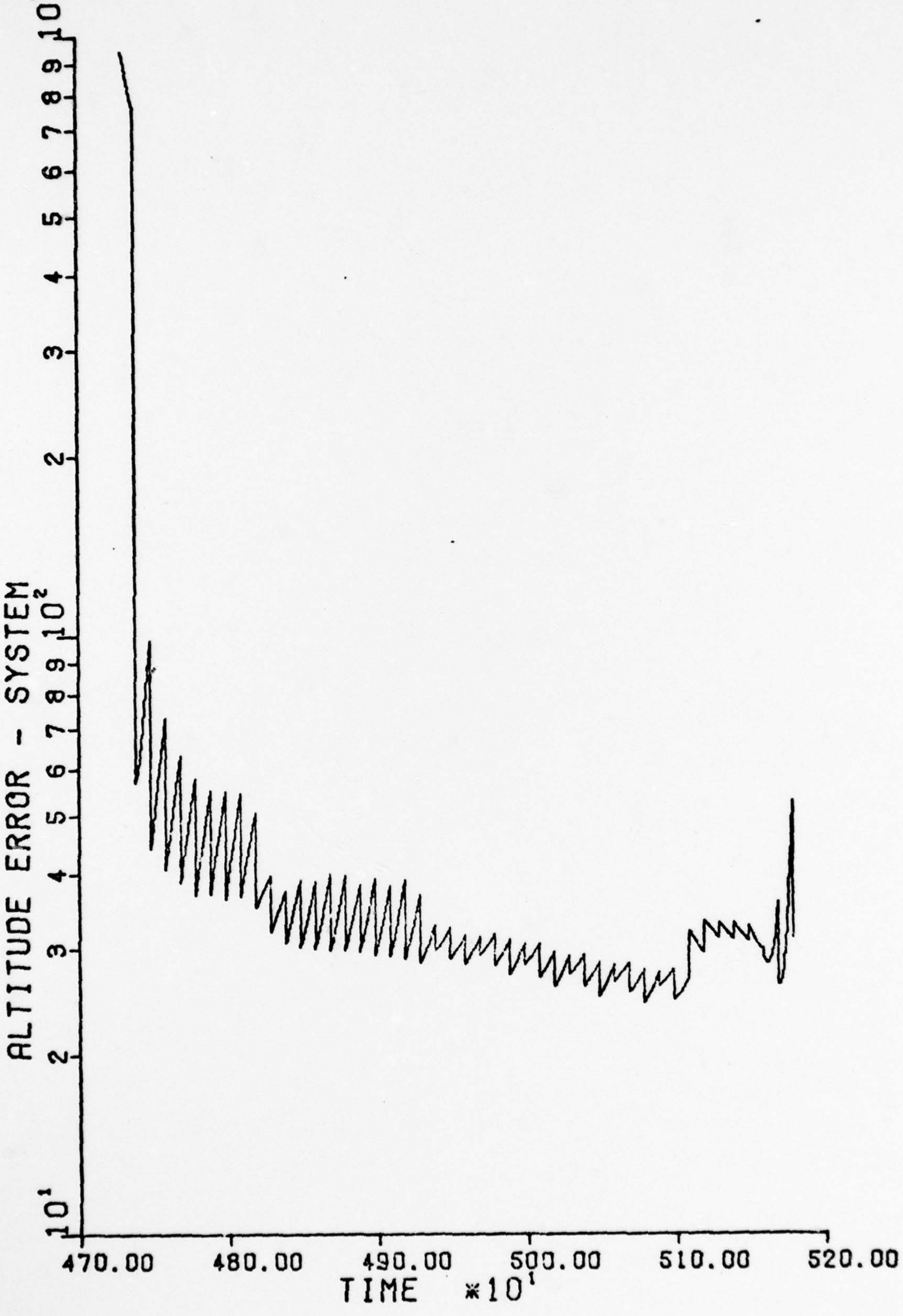


Figure 42. Filter IV True Altitude Error

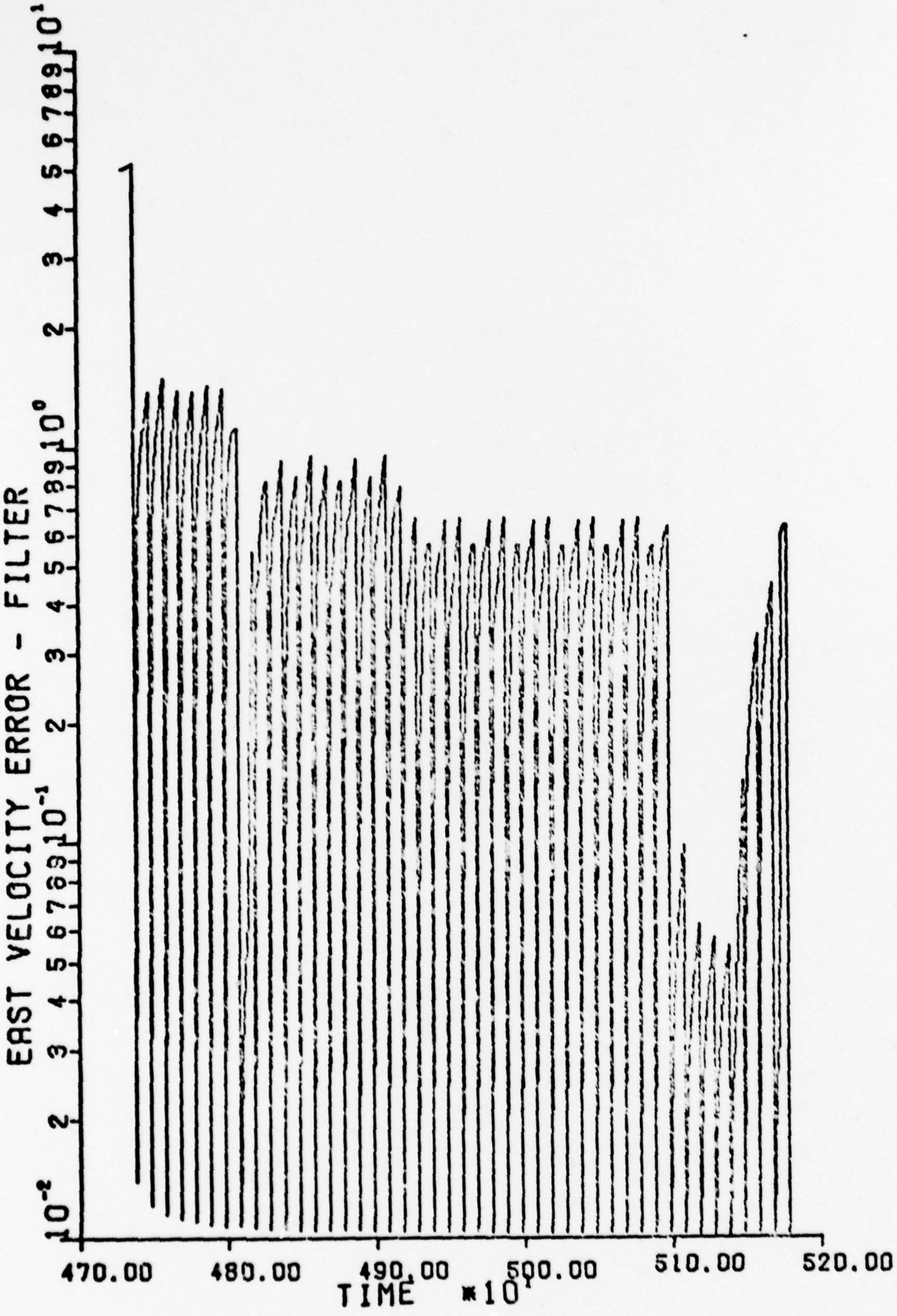


Figure 43. Estimated East Velocity Error

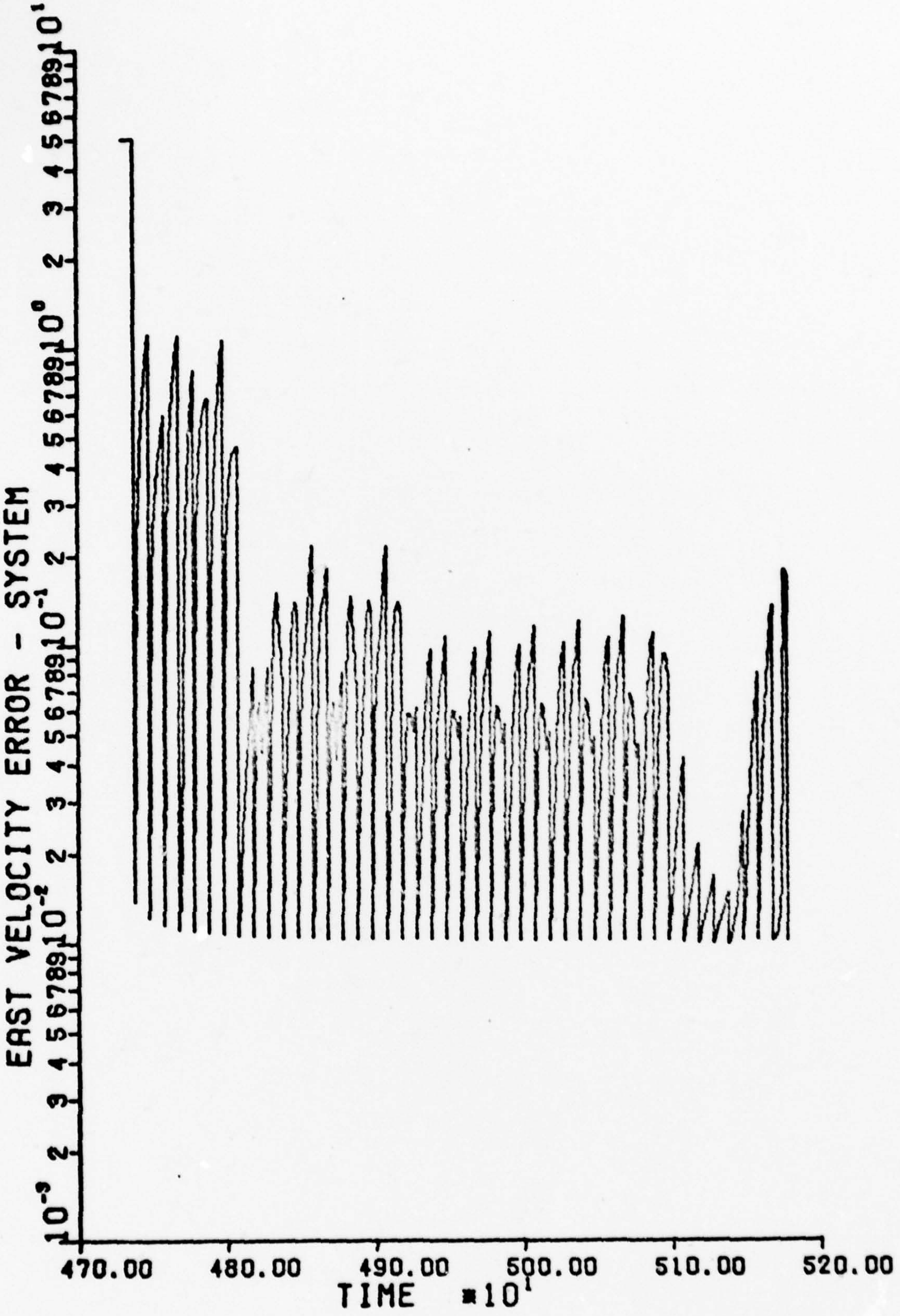


Figure 44. Filter IV True East Velocity Error

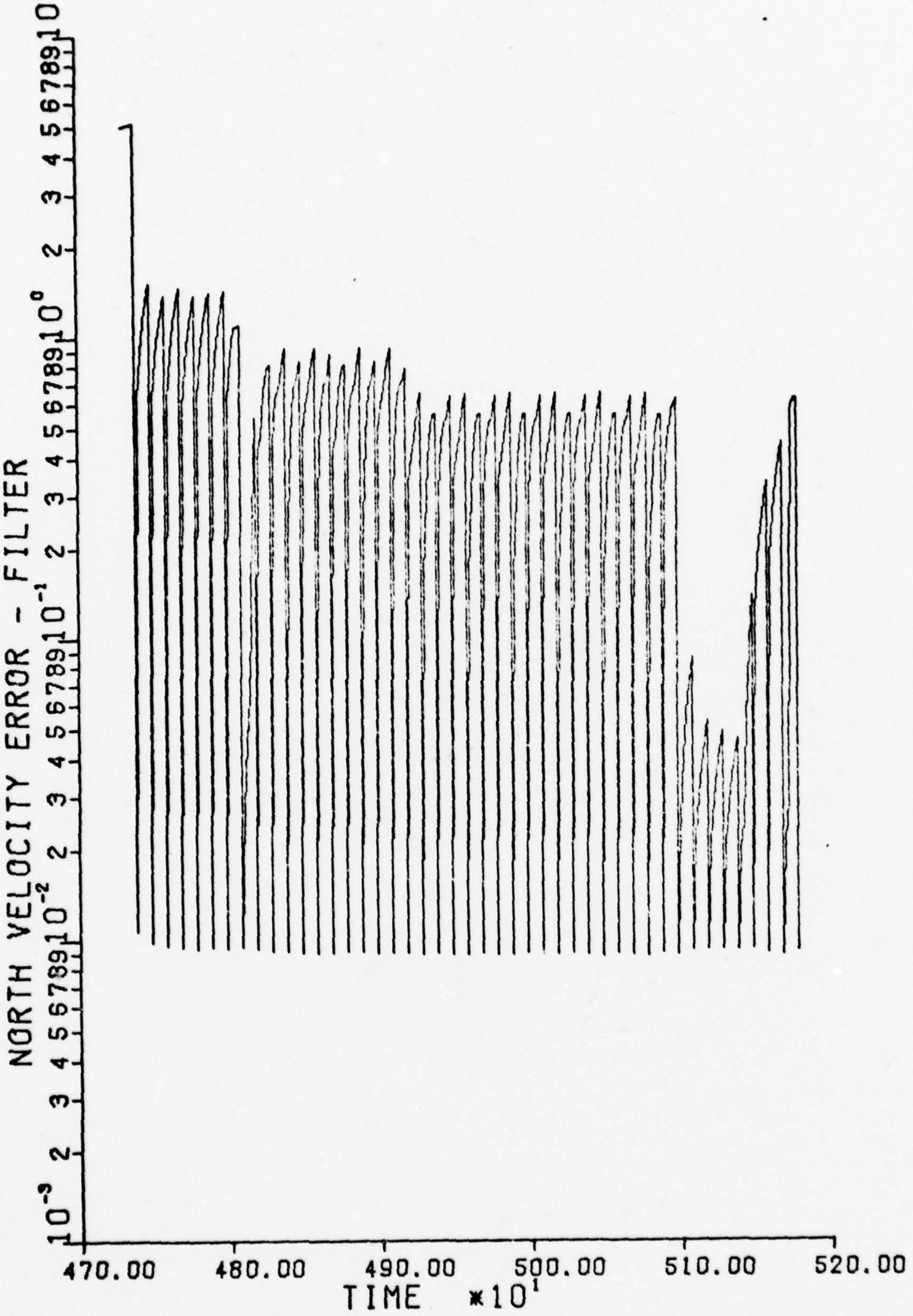


Figure 45. Filter IV Estimated North Velocity Error

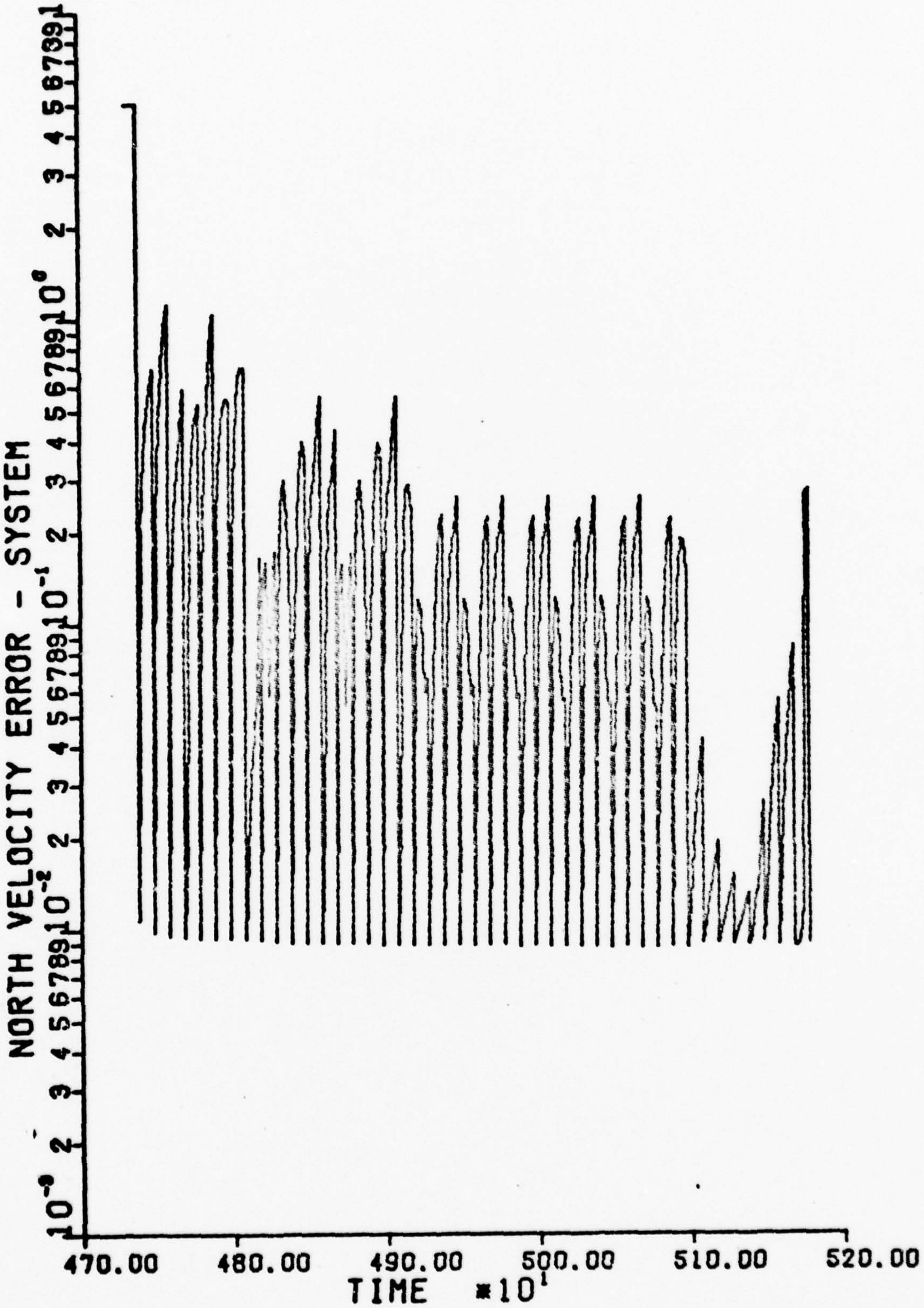


Figure 46. Filter IV True North Velocity Error

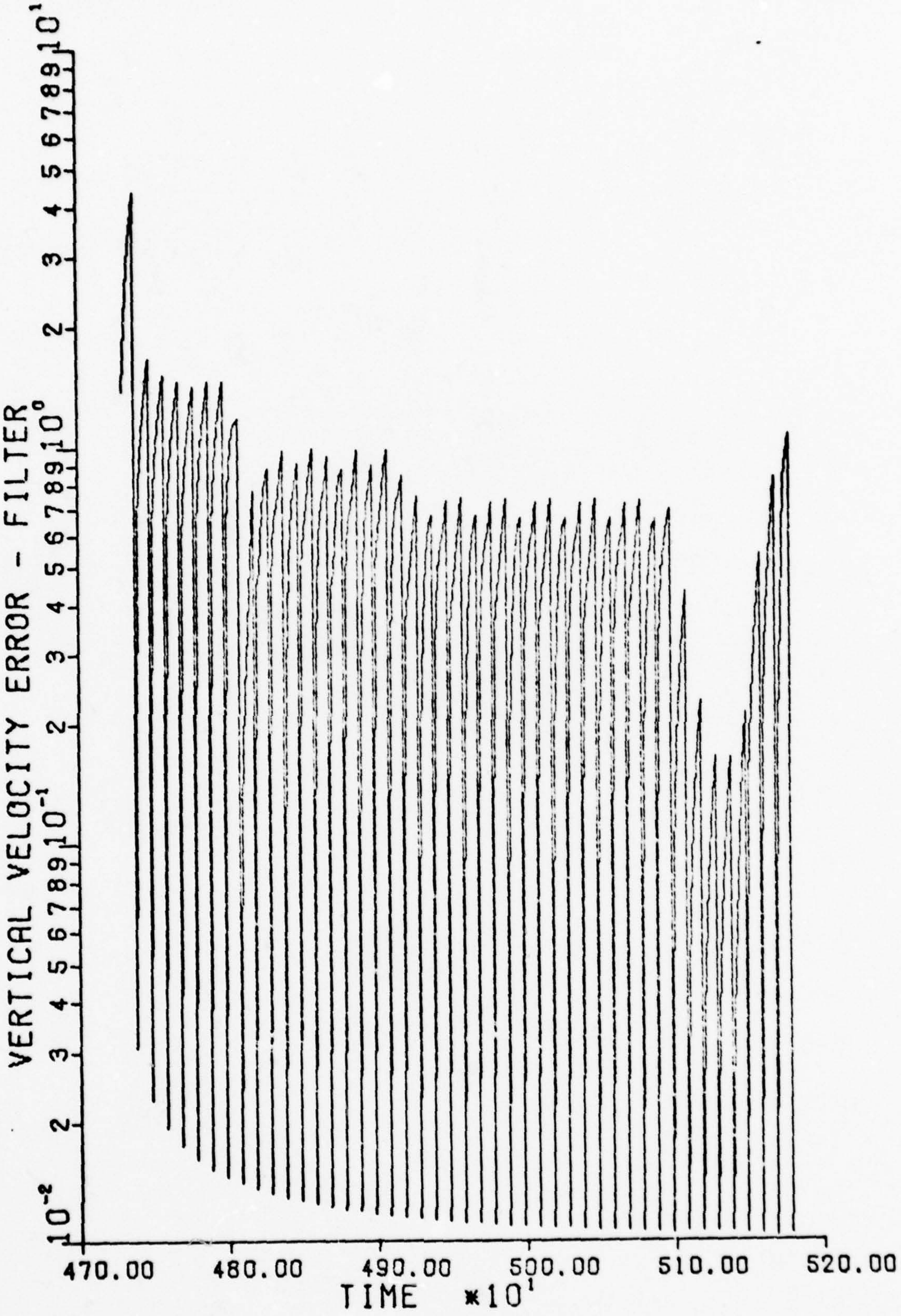


Figure 47. Filter IV Estimated Vertical Velocity Error

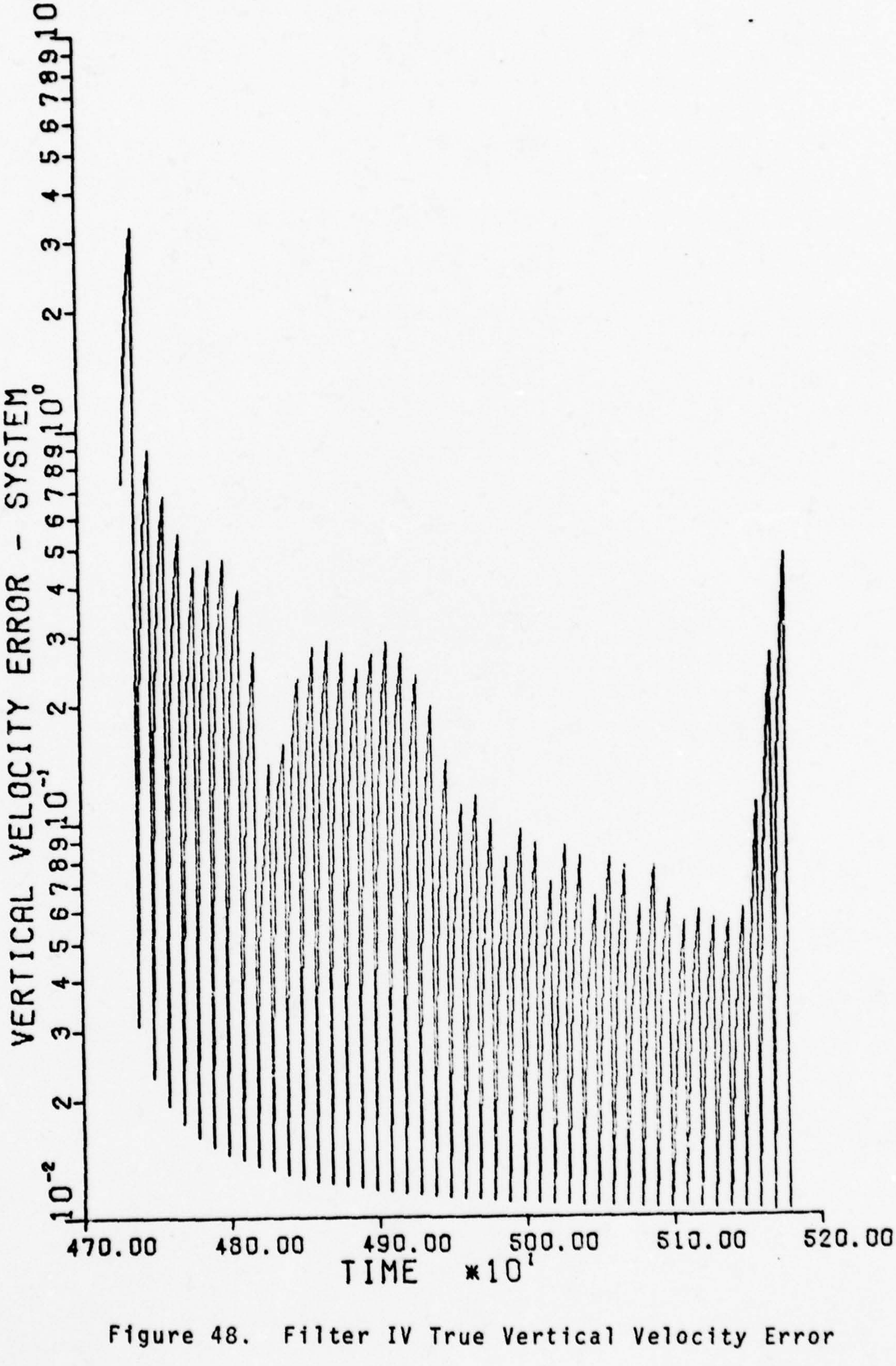


Figure 48. Filter IV True Vertical Velocity Error

minimize these errors. Therefore, a smaller value of the "SYSTEM" rms errors indicates better filter performance.

The data shown in Figures 1 to 48 indicates that there is very little difference in the performances of the four filters tested. Most of the plotted lines are so similar that it is difficult to distinguish any difference at all. In testing, they can be compared by placing one directly over the other, with the axes aligned. Since that method is not practical here, the actual numbers plotted for a typical post-transient time point are listed in Table VIII.

The data produced with an integration stepsize of 0.2 seconds was so close to the values computed for a stepsize of 2 seconds that the plots are indistinguishable. These error values are listed in Table IX. Because the plots are so similar to Figures 1 to 48, they are not shown.

Discussion of Results

Figures 1 to 48 and Table VIII show that there is virtually no difference in the performances of the proposed filters. The only significant differences appear in the vertical channels (altitude and vertical velocity errors). However, the vertical channel model has deficiencies which place these results in doubt. Myers and Butler (Ref 2) reported that the vertical channel model used for this study diverged from correct state estimation for a long flight. For the shorter flight profile of this study, divergence did not occur. However, the vertical channel model is not considered suitable for an accurate measurement of Kalman

Table VIII

Steady State RMS Estimation Errors $\Delta t = 2.0$ Seconds

Component		Type I	Type II	Type III	Type IV
Longitude	-	3.58	4.32	3.42	3.92
(Radiansx10^{-7})	+	3.53	4.29	3.41	3.83
Latitude	-	2.67	2.70	2.94	2.76
(Radiansx10^{-7})	+	2.51	2.55	2.50	2.59
Altitude	-	47.9	50.6	50.4	27.8
(Feet)	+	39.9	42.9	35.7	24.7
East Velocity	-	.054	.056	.029	.078
(Feet/sec)	+	.010	.010	.010	.010
North Velocity	-	.136	.138	.380	.145
(Feet/sec)	+	.010	.010	.009	.009
Vertical Velocity	-	.084	.262	.349	.063
(Feet/sec)	+	.011	.011	.011	.010

NOTES:

- Indicates the error prior to measurement incorporation.

+ Indicates the error after measurement incorporation.

Errors listed are for time 5098 seconds. This time was selected as representative of steady state performance.

Longitude: 10^{-7} radians = 1.54 feet (at Lat=42.4°)

Latitude: 10^{-7} radians = 2.09 feet

Table IX

Steady State RMS Estimation Errors $\Delta t = 0.2$ seconds

Component		Type I	Type II	Type III	Type IV
Longitude	-	3.56	4.20	3.41	3.91
(Radians $\times 10^{-7}$)	+	3.52	4.18	3.41	3.83
Latitude	-	2.68	2.71	2.92	2.76
(Radians $\times 10^{-7}$)	+	2.51	2.55	2.50	2.60
Altitude	-	48.0	52.0	50.5	26.8
(Feet)	+	39.6	42.3	35.6	24.5
East Velocity	-	.052	.054	.023	.075
(Feet/sec)	+	.010	.010	.010	.010
North Velocity	-	.130	.136	.376	.139
(Feet/sec)	+	.010	.010	.009	.009
Vertical Velocity	-	.076	.102	.324	.062
(Feet/sec)	+	.011	.011	.011	.010

See notes for Table VIII.

Filter performance, so the vertical channel results were not used for comparison of the proposed filters. The other variables errors in East and North components of position and velocity, indicate that the filters are achieving almost identical performance. This appears to contradict the expectation of a performance gain suggested by the development of Chapters II and III. Several factors which could cause this result must be considered.

One factor is that the $\underline{Q}_d$ matrix is only one of the parts of the Kalman Filter model which significantly affect estimation accuracy. From equation (17), it is seen that the covariance which the filter estimates and uses to compute the gain matrix is driven by the state transition matrix, $\Phi(t_{i+1}, t_i)$, as well as the driving noise matrix. If the driving noise is small, the first term in (17) could dominate the computation. Then almost any form of a $\underline{Q}_d$ matrix with reasonably accurate tuning would produce similar results in this testing. It might further be expected that if $\underline{Q}_d$ is small compared to $\underline{\Phi}\underline{P}\underline{\Phi}^T$, making the $\underline{Q}_d$ terms even smaller would have little effect on performance. However, it was found during the tuning process that significant reduction in the $\underline{Q}_d$ terms caused a severe performance degradation. Therefore, the idea that the similarity of the test results stem from a lack of filter performance sensitivity to the proper $\underline{Q}_d$ values can be rejected.

Another factor is that the filters were not exhaustively fine-tuned, as mentioned in Chapter I. All of the filters were tuned until further tuning efforts appeared to have only marginal and somewhat ambiguous effects. However, there is still the possibility that futher refinements in tuning would ultimately give one of the filters significantly better performance than the others. The tuning effort was extensive enough (approximately 120 tuning runs) that this possibility is judged to be unlikely.

A factor that appears to be significant in these results is the accuracy of the measurements available from the GPS receiver. In the model used (Appendix A), these measurements are modeled as range and range rate measurements from each of four satellites. The one-sigma errors of these measurements are 20 feet for range and 0.1 foot/sec. for range rate. The combined information from four such satellites can yield measurements of even greater accuracy, depending on the geometry of the satellites when a measurement is taken. For example, a single observation of all four satellites measure position with an error as small as

$$\frac{1}{4\sqrt{\left(\frac{1}{20\text{ ft}}\right)^2}} = 10 \text{ feet}$$

As long as the Kalman Filter continues to run without divergence (i.e., without reducing the gain to ignore external measurements), these highly accurate measurements will allow it to have small errors in position and velocity. Even when the gain is substantially different from the optimal value, the errors will be small. Thus, the effect of a better Q_d matrix in more accurately calculating the gain matrix produces only a very small benefit in reducing the errors in the filter.

As mentioned in Chapter IV, for a small integration stepsize, the off-diagonal terms in Q_d are insignificant in comparison to the diagonal terms. It is possible that for an integration stepsize of 2 seconds, these off-diagonal terms are still not extremely significant, and thus that they make little difference in fiter performance. Also, in the Type III Q_d matrix, the off-diagonal terms are computed directly from the noise values of the Q matrix in the truth model (see Table IV). Then pseudo-noises are added to the diagonal terms. These pseudo-noises may be substantially larger than the true noise terms from which the correlated off-diagonal terms were computed. Thus, the off-diagonal terms in this Q_d matrix could be disproportionately low, so that they have little effect on performance.

Conversely, the off-diagonal terms for the Type II Q_d matrix were computed with the correlation factors from the Type III analysis, but using the pseudo-noise values of the diagonal terms. This could possibly make the off-diagonal terms too large, i.e., it might be more appropriate to have only some of the pseudo-noise appear in the correlated terms. Provisions were made in the covariance analysis program to scale the off-diagonal terms by an arbitrary factor, in order to test this idea. However, this testing was not performed due to time limitations. Thus, the possibility that the off-diagonal terms as computed were too large or too small to give proper results must be considered.

All of the potential reasons given so far for the test results seen in Figures 1 to 48 and Tables VIII and IX are at least possible, and some must be considered highly probable. However, there is another factor which outweighs all of them. This effect is caused by the way in which the time propagation equation for the covariance is computed (equation (23)) is used in GCAP, but equation (12) could be used instead). The computation used in the testing for this study included a partitioning of the integration interval, which tended to mask the effects of off-diagonal terms.

To see this, consider how $\underline{P}(t)$ is propagated. As explained in Chapter IV, equation (23) cannot be used directly, as GCAP would normally do. Instead, $\underline{Q}(t)$ is set to zero, equation (23) is integrated, and finally $\underline{Q}_d$ is added:

$$\underline{P}^-(t+\Delta t)=\int_t^{t+\Delta t}[\underline{F}(\tau)\underline{P}(\tau)+\underline{P}(\tau)\underline{F}^T(\tau)]d\tau+\underline{Q}_d(t+\Delta t) \qquad (24)$$

Theoretically, this computation should be performed once for the interval from one update to the next. However, in GCAP a Runge-Kutta integration technique is used, which is not sufficiently accurate with a stepsize equal to the typical update interval of approximately 10 seconds. In most on-line Kalman Filter applications, an even simpler Euler integration is used to evaluate equation (24) or an equivalent form such as equation (12). Thus, to achieve sufficient numerical accuracy, the integration stepsize must be reduced. However, there is no need to reduce the update interval.

An integration stepsize smaller than the update inerval is often called an integration sub-interval. Using this idea, equation (24) is evaluated for some Δt, smaller than the 10 second measurement update interval, which gives accurate numerical integration. This caluclation is repeated until the next update time is reached, then a measurement is processed. In the testing for this study, the update interval was set at 10 seconds. This was divided into 5 sub-intervals of 2 seconds for propagation via equation (24).

In this type of propagation, the $\underline{Q}_d$ matrix will be added to intermediate values of $\underline{P}$. Then these $\underline{P}$ matrices will be used in the integral term of equation (24). If the $\underline{Q}_d$ matrix is assumed to be diagonal, the diagonal elements of the intermediate $\underline{P}$ matrices will be directly affected by the $\underline{Q}_d$ terms. The products $\underline{F}\underline{P}$ and $\underline{P}\underline{F}^T$ will generate off-diagonal terms from the diagonal elements of $\underline{P}$. Then the integral term of equation (24) will compute off-diagonal terms in the propagated $\underline{P}$ matrix based on the diagonal elements of $\underline{Q}_d$. Thus, the effect of a diagonal form of $\underline{Q}_d$ added at each sub-interval is equivalent to the effect of a full $\underline{Q}_d$ matrix (i.e., containing off-diagonal terms) added only at the last sub-interval before a measurement incorporation.

This effect of producing off-diagonal terms in $\underline{P}$ from a diagonal $\underline{Q}_d$ by the nature of the integration used appears to be the most significant factor in the lack of any performance variation among the filters tested.

Conclusion

The use of sub-intervals for covariance propagation in Kalman Filters is a common technique. The results of this study indicate that this technique is appropriate in many cases. Under the conditions that

1) accurate measurements are available to the navigation system
2) several integration sub-intervals are used for covariance propagation,
3) a properly tuned, diagonal form of $\underline{Q}_d$ is added for each sub-interval,

the testing showed that the estimation results are comparable to the results obtainable from larger, more complex forms of the $\underline{Q}_d$ matrix. Since most Kalman Filters must use the sub-interval technique for covariance propagation to achieve sufficient numerical accuracy, the use of a computationally advantageous diagonal state noise covariance matrix is justified.

Recommendations

The alternate forms of $\underline{Q}_d$ derived for this study could still have some applicability which warrants futher investigation. Further testing should be performed with these forms to evaluate the effects of what appear to be the major factors in causing the observed test results. In order to perform this testing, it is recommended that the new forms of the truth model and filter model designed by Intermetrics Incorporated (Ref 6) be used. This would allow analysis of vertical channel performance, which appears to be more sensitive to variations in the Kalman Filter than the horizontal channels.

The basic variation in this testing would be a change in the method used for covariance propagation. The GCAP subroutine which performs this function (INTEG) was modified for this study to perform the evaluation as in equation (24) rather than (23). This change involved eliminating the addition of $\underline{Q}(t)$ in the $\dot{\underline{P}}(t)$ equation before integration, and adding a subroutine called (INTQ) to add $\underline{Q}_d(t)$ to the propagated $\underline{P}(t)$ matrix. This can easily be modified to add $\underline{Q}_d(t)$ to $\underline{P}(t)$ only for the last integration step before a measurement update. In order to evaluate the $\underline{Q}_d$ matrix, a value of Δt is needed. In the testing performed, Δt was set equal to the integration stepsize. For the proposed tests, Δt would have to be set to the update interval, since the $\underline{Q}_d$ matrix must represent the total noise contribution for this interval.

These changes to the testing configuration will almost certainly require retuning of the proposed Kalman Filters. It is recommended that the filters be retuned and tested with these changes in the models and covariance propagation equations, but with the other test parameters the same as in this study. Also, a filter should be tested with a diagonal $\underline{Q}_d$ added for each sub-interval, as Type I in this study. This would provide a comparison to show whether the proposed filters, with $\underline{Q}_d$ added only once for the total propagation, could give performance equal to that of a standard filter, with a diagonal $\underline{Q}_d$ added for each sub-interval.

This comparison should indicate whether equivalent performance can be obtained with a less frequent addition of a full $\underline{Q}_d$ matrix, rather than a frequent addition of a diagonal $\underline{Q}_d$ matrix. If this is true, it could lead to a significant savings in computations for an on-line Kalman Filter. For a typical sub-interval size of 0.2 seconds, the 16 term diagonal $\underline{Q}_d$ matrix is added 50 times, for a total of 800 additions for the $\underline{Q}_d$ term of the covariance propagation between updates. With a $\underline{Q}_d$ matrix containing off-diagonal terms added once per measurement sample period, such as Type II in Chapter III, this could be reduced to 30 additions, while adding only 7 extra words of storage space. This could be a substantial savings in computation.

The effect of less accurate measurements can be tested in a very straight forward way by simply increasing the magnitude of the terms in the measurement covariance matrix, $\underline{R}$. With this change, the performance of the Kalman Filter would rely more heavily on the accuracy of the internal model. Thus, for less accurate measurements, a better evaluation of the $\underline{Q}_d$ matrix would yield a more substantial improvement in filter performance than is seen when highly accurate external measurements are available.

The off-diagonal terms generated by sub-interval propagation of a diagonal $\underline{Q}_d$ matrix are not the same as the off-diagonal terms in the Type III $\underline{Q}_d$ matrix. The Type III matrix contains relatively few off-diagonal terms because it is based on the $\underline{Q}$ matrix from the truth model. This $\underline{Q}$ matrix has only four noise terms directly driving the states that are included in the filter model (states 1-16). When equation (24) is used for covariance propagation, all 16 states have driving pseudo-noises, so many more off-diagonal terms will be generated. Another $\underline{Q}_d$ matrix could be derived in the same way that the Type III matrix was derived, but with the pseudo-noises included in states 1-16 of the $\underline{Q}$ matrix when equation (20) is evaluated. The filter based on this type of $\underline{Q}_d$ matrix could then be compared with one in which only true driving noises are used in computing off-diagonal terms, as in Type III.

Other parameters could also be tested. The proposed filters could be tested using a long, low-dynamics flight profile, in which gyro drifts, Schuler oscillations, and other long term effects could cause problems. The effect of multiplying the off-diagonal terms in the derived $\underline{Q}_d$ matrices by a scale factor for improved tuning could be evaluated. All of this testing could lead to a more accurate $\underline{Q}_d$ matrix, which could reduce the computational burden of the Kalman Filter without reducing navigational accuracy.

A limited amount of additional testing suggested in this section was attempted. The only change made for this testing was to add $\underline{Q}_d$ only for the last integration sug-interval before an update. Runs were made using the Type II $\underline{Q}_d$ matrix, with all the terms in Table VI multiplied by 5 to account for the longer interval for which the pseudo-noises simulated the effects of real errors. To obtain good performance, this filter would need to be retuned, but time limitations prevented this.

The most noticeable change in the results in this test was in the filter covariance propagation. With $\underline{Q}_d$ added for each sub-interval, the covariance grows quickly. With $\underline{Q}_d$ added only for the last sub-interval, the covariance grows more slowly, then has a step increase at the end of the propagation. This allows the effects of the homogeneous covariance growth and the added covariance to be distinguished in the test results.

To illustrate this, Figures 49 and 50 show an expanded scale for the filter estimates of north velocity error. In Figure 49, $\underline{Q}_d$ was added for each sub-interval. In Figure 50, $\underline{Q}_d$ was added only for the last sub-interval. Note that the plotter interpolates between data points. The dotted lines in Figure 50 show how the covariance would actually propagate for small integration steps. This would be followed by a step rise in the covariance when $\underline{Q}_d$ is added.

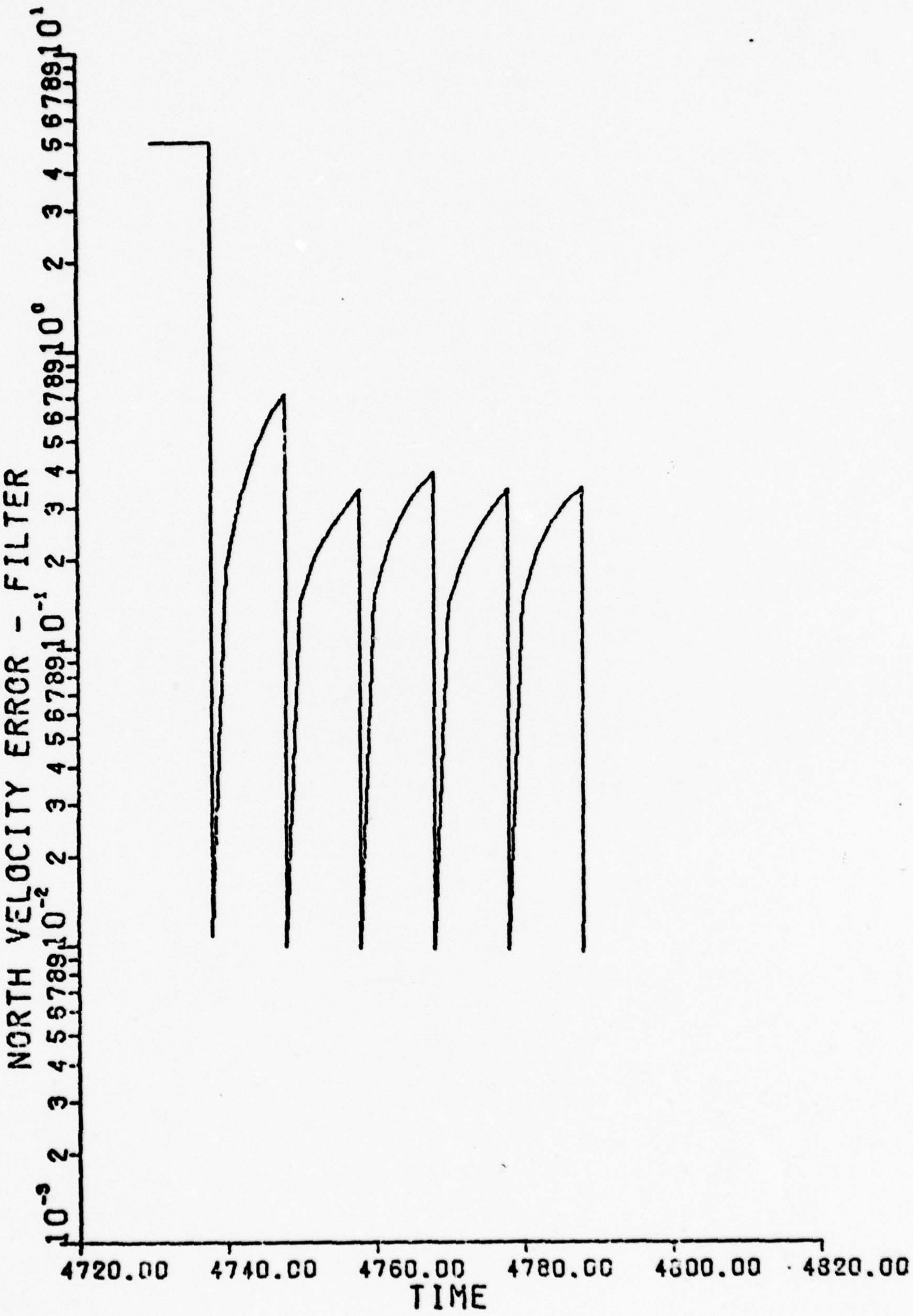


Figure 49. Error Growth with $\underline{Q}_d$ Added for Each Sub-Interval

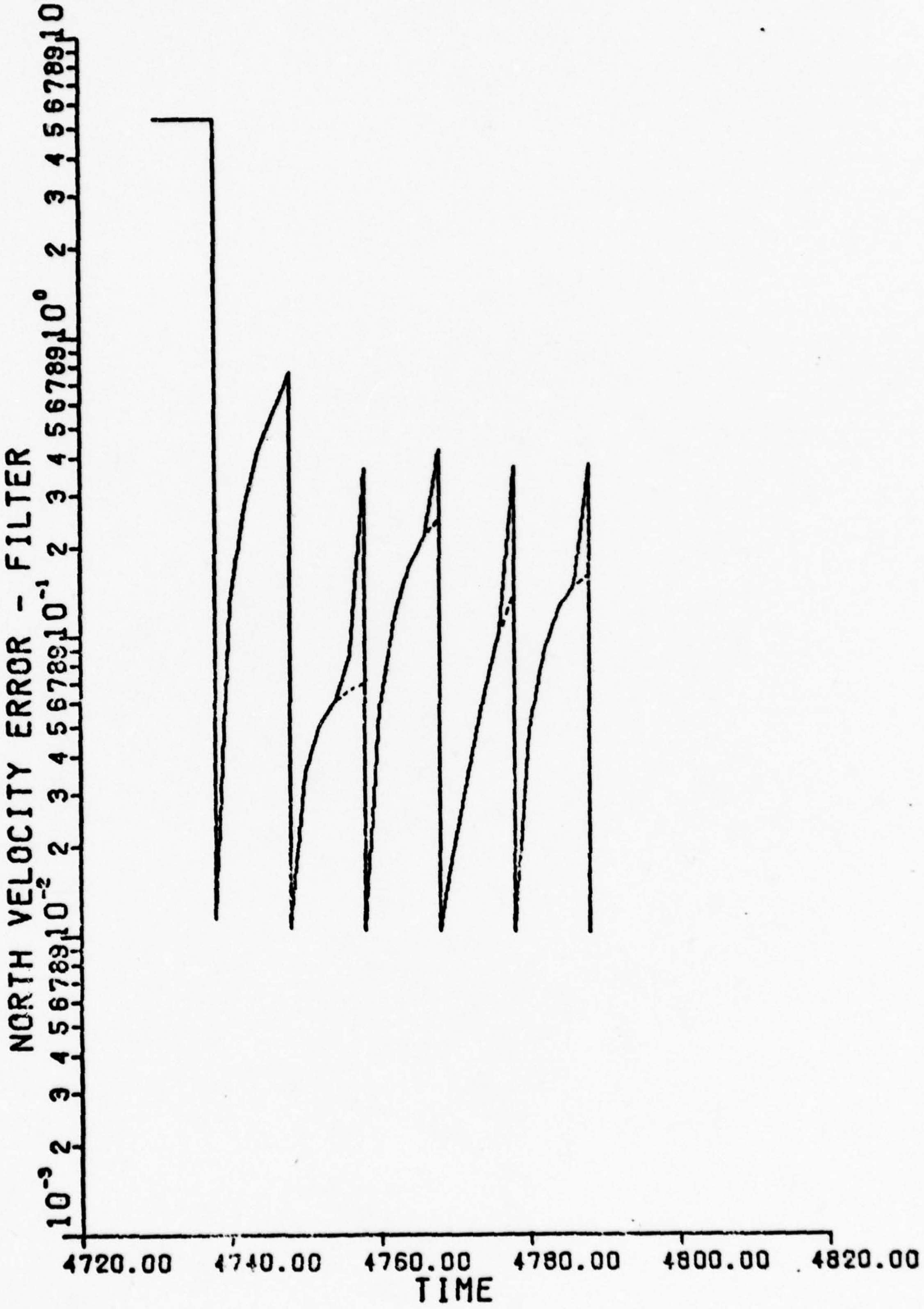


Figure 50. Error Growth with $\underline{Q}_d$ Added Only for Last Sub-Interval

If $\underline{Q}_d$ is added only for the last sub-interval, off-diagonal terms must be included in $\underline{Q}_d$ to simulate the off-diagonal terms generated by a proper evaluation of equation (23). A Kalman Filter using such a $\underline{Q}_d$ matrix shows promise of giving good performance with slightly more storage space and considerably less computation time than the usual form requires.

Summary

The test results presented in this chapter are inconclusive in demonstrating the advantages of the proposed state noise covariance matrices. They indicate that, for the type of sub-interval propagation equations commonly used in Kalman Filters, the usual diagonal form of $\underline{Q}_d$ provides performance comparable to that obtained with larger, more complex forms of $\underline{Q}_d$. Futher testing could result in an improved means of incorporating driving noise into the covariance propagation. This method would involve a $\underline{Q}_d$ matrix containing off-diagonal terms, which would be added only once for a covariance propagation. Thus, it would significantly reduce the number of filter computations required, with a modest increase in computer storage space. This change could be found to have little or no adverse effect on filter estimation performance. Moreover, alternate forms of $\underline{Q}_d$ could be found to yield significantly enhanced filter performances for applications in which the external measurements are not of the extremely high accuracy characteristic of GPS measurements.

Thus, for many practical Kalman Filter applications, the ideas of the alternate forms of discrete time state noise covariance matrices discussed in this study appear to be a fruitful area for continued investigation.

Bibliography

1. Widnall, W. S. and P. A. Grundy. Inertial Navigation System Error Models. Cambridge, Mass.: Intermetrics, Incorporated, 11 May 1973.

2. Myers, K. A. and R. R. Butler. "Simulation Results for an Integrated GPS/Inertial Aircraft Navigation System." Proceedings of the IEEE National Aerospace and Electronics Conference, NAECON '76. New York: Institute of Electrical and Electronics Engineers, Incorporated, 1976.

3. Widnall, W. S. and P. K. Sinha. Comparison of Three Vertical Channel Designs for an Integrated GPS/Inertial Navigation System. Cambridge, Mass.: Intermetrics, Incorporated, 27 July 1977.

4. Maybeck, Peter S. Stochastic Models, Estimation and Control. Textbook manuscript, 1975.

5. Widnall, W. S., N. A. Carlson and P. A. Grundy. Post Flight Processor for CIRIS. Cambridge, Mass.: Intermetrics, Incorporated, 24 November 1972.

6. Sinha, P. K. Integrated GPS/Inertial Simulator Computer Program. Cambridge, Mass.: Intermetrics, Incorporated, 4 August 1977.

7. Hamilton, E. L., G. Chitwood and R. M. Reeves. The General Covariance Analysis Program (GCAP), An Efficient Implementation of the Covariance Analysis Equations. Preliminary copy, Wright-Patterson Air Force Base, Ohio: Air Force Base, Ohio: Air Force Avionics Laboratory, 1976.

8. Musick, Stanton H. PROFGEN - A Computer Program for Generating Flight Profiles. Wright-Patterson Air Force Base, Ohio: Air Force Avionics Laboratory, March 1976.

Appendix A

Integrated GPS/Inertial Navigation System Models

This section describes the truth model for an integrated GPS/Inertial navigation system. It also describes the reduced order model on which a Kalman Filter is based. These models were used in the testing for this thesis. The basic truth model is a 48-state model of a baro-inertial navigation system using a representative inertial navigation unit in the 1 nautical mile/hour class. This model was derived by Widnall and Grundy (Ref 1). Myers and Butler of the Air Force Avionics Laboratory modified this model by adding the necessary user clock states to represent a typical Global Positioning System (GPS) receiver (Ref 2). The truth model contains 52 states, of which the first 16 are included in the filter model.

The models are described in terms of four matrices: $\underline{F}$, $\underline{Q}$, $\underline{H}$, and $\underline{R}$. The $\underline{F}$ matrix is derived from the nonlinear homogeneous state differential equation for the system, as in equation (2). $\underline{F}$ is then a linearized matrix describing the homogeneous error state behavior. $\underline{Q}$ is the matrix of the strengths of the continuous time white, Gaussian noises which drive the system. $\underline{H}$ is an observation geometry matrix, as used in equation (4). $\underline{R}$ is the measurement covariance matrix, which describes the accuracy of the measurement information matrix.

The $\underline{H}$ matrix used in testing is for a single instance of receiver and satellite geometry. As explained in Chapter

IV, the observation geometry was assumed to remain constant for the test runs in this study, although in reality this matrix would change as the aircraft and satellites moved. This $\underline{H}$ matrix was generated by randomly selecting aircraft and satellite positions and using a satellite selection routine from the Integrated GPS/Inertial Simulation Program (Ref 6). The $\underline{H}$ matrix entries are identical for the truth model and filter model, except for the extra zeroes added to the truth model to obtain the proper dimension.

The $\underline{F}$, $\underline{Q}$, $\underline{H}$, and $\underline{R}$ matrices are listed in Tables X to XIII. Distinctions between the truth model and filter model matrices are described in these tables. The state variables for the filter and truth models are listed in Tables I and II of Chapter III.

Table X a

$\underline{F}$ Matrix - Notation

α = Wander azimuth angle (radians)

Lat = Vehicle Latitude (radians)

Lon = Vehicle Longitude (radians)

h = Vehicle altitude (feet)

V = Vehicle velocity (feet/sec). Components in E-N-Z (East-North-Up) frame shown as V_E, V_N, V_Z

F = Specific force vector. Components in wander azimuth frame are F_s, F_y, F_z. In the E-N-Z frame,

$$F_E = -F_x \cdot \sin(\alpha) - F_y \cos\cdot(\alpha)$$

$$F_N = F_x \cdot \cos(\alpha) - F_y \sin\cdot(\alpha)$$

R = Earth equatorial radius (feet)

ρ = Angular velocity of local E-N-Z coordinates with respect to earth (rad/sec)

$$\rho_E = -V_N/R$$

$$\rho_N = V_E/R$$

$$\rho_Z = V_E \cdot \tan(Lat)/R$$

Ω = Earth angular rate (rad/sec). In local coordinates:

$$\Omega_N = \Omega \cdot \cos(Lat)$$

$$\Omega_Z = \Omega \cdot \sin(Lat)$$

ω = Angular velocity of local E-N-Z coordinates with respect to inertial space (rad/sec). Components are

$$\omega_E = \rho_E$$

$$\omega_N = \rho_N + \Omega_N$$

$$\omega_Z = \rho_Z + \Omega_Z$$

$$\omega_X = \omega_e \cdot \cos(\alpha) + \omega_N \cdot \sin(\alpha)$$

$$\omega_Y = -\omega_E \cdot \sin(\alpha) + \omega_N \cdot \cos(\alpha)$$

k_1, k_2, k_3 = Damping coefficients for baro-inertial altitude channel

D_{GE}, D_{GN}, D_{GZ} = Correlation distances of gravity deflections (feet)

D_{ALT} = Correlation distance of altimeter error (feet)

β_{tru} = Inverse correlation time of clock random frequency error (1/sec)

G_Z = Vertical component of gravity (ft/sec^2)

Table X b

$\underline{F}$ Matrix - Position Errors

Longitude Error

$F(1,2) = \rho_z/\cos(Lat)$

$F(1,3) = -\rho_N/(R \cdot \cos(Lat))$

$F(1,4) = 1/(R \cdot \cos(Lat))$

Latitude Error

$F(2,3) = \rho_E/R$

$F(2,5) = 1/R$

Altitude Error

$F(3,3) = -k_1$

$F(3,6) = 1$

$F(3,10) = k_1 \cdot h$

$F(3,47) = k_1$

Table X c
$\underline{F}$ Matrix - Horizontal Velocity Errors

East Velocity Error
F(4,2) = $2\cdot(\Omega_N\cdot V_N+\Omega_Z\cdot V_Z)+\rho_N\cdot V_N/(\cos(Lat))^2$
F(4,3) - $\rho_Z\cdot\rho_W+\rho_N\cdot V_Z/R)$
F(4,4) = $-(\rho_E\ \tan(Lat)+V_Z/R)$
F(4,5) = $2\cdot\Omega_Z+\rho_Z$
F(4,6) = $-(2\cdot\Omega_N+\rho_N)$
F(4,8) = $-F_Z$
F(4,9) = F_N
F(4,35) = $\cos(\alpha)$
F(4,36) = $-\sin(\alpha)$
F(4,38) = $F_X\cdot\cos(\alpha)$
F(4,39) = $-F_y\cdot\sin(\alpha)$
F(4,41) = $-(F_A+G_A)\cdot\cos(\alpha)$
F(4,42) = $F_y\cdot\cos(\alpha)$
F(4,43) = $-(F_A+G_A)\cdot\sin(\alpha)$
F(4,44) = $f_x\cdot\sin(\alpha)$
F(4,48) = 1

North Velocity Error
F(5,2) = $-(2\cdot\Omega_N\cdot V_E+\rho_N\cdot V_E/(\cos(Lat))^2)$
F(5,3) = $\rho_N\cdot\rho_Z-\rho_E\cdot V_Z/R$
F(5,4) = $-2\cdot\omega_z$
F(5,5) = V_Z/R
F(5,6) = ρ_E
F(5,7) = F_Z
F(5,9) = $-F_E$
F(5,35) = $\sin(\alpha)$
F(5,36) = $\cos(\alpha)$
F(5,38) = $F_x\cdot\sin(\alpha)$
F(5,39) = $F_y\cdot\cos(\alpha)$
F(5,41) = $-(F_Z+G_Z)\cdot\sin(\alpha)$
F(5,42) = $F_y\cdot\sin(\alpha)$
F(5,43) = $(F_Z+G_Z)\cdot\cos(\alpha)$
F(5,44) = $-F_x\cdot\cos(\alpha)$
F(5,49) = 1

Table X d

$\underline{F}$ Matrix - Vertical Velocity Error

Vertical Velocity Error

$F(6,2) = -2 \cdot \Omega_Z \cdot V_E$

$F(6,3) = 2 \cdot G_Z / R - xk2 - (\rho_E^2 + \rho_N^2)$

$F(6,4) = 2 \cdot \omega_N$

$F(6,5) = -2 \cdot \rho_E$

$F(6,7) = -F_N$

$F(6,8) = F_E$

$F(6,10) = k_2 \cdot h$

$F(6,11) = -1$

$F(6,37) = 1$

$F(6,40) = F_Z + G_Z$

$F(6,45) = -F_y$

$F(6,46) = F_x$

$F(6,47) = k_2$

$F(6,50) = 1$

Table X e
$\underline{F}$ Matrix - Tilt Errors

East Axis Tilt

$F(7,3) = -\rho_E/R$
$F(7,5) = -1/R$
$F(7,8) = \omega_Z$
$F(7,9) = -\omega_N$
$F(7,14) = \cos(\alpha)$
$F(7,15) = -\sin(\alpha)$
$F(7,17) = F_x \cdot \cos(\alpha)$
$F(7,18) = F_y \cdot \cos(\alpha)$
$F(7,19) = -F_y \cdot \sin(\alpha)$
$F(7,20) = -F_y \cdot \sin(\alpha)$
$F(7,23) = F_x \cdot F_y \cdot \cos(\alpha)$
$F(7,24) = -F_x \cdot F_y \cdot \sin(\alpha)$
$F(7,26) = \omega_x \cdot \cos(\alpha)$
$F(7,27) = -\omega_y \cdot \sin(\alpha)$
$F(7,29) = \Omega_z \cdot \cos(\alpha)$
$F(7,30) = -\omega_y \cdot \cos(\alpha)$
$F(7,31) = \Omega_z \cdot \sin(\alpha)$
$F(7,32) = -\omega_x \cdot \sin(\alpha)$

North Axis Tilt

$F(8,2) - -\Omega_z$
$F(8,3) = -\rho_N/R$
$F(8,4) = 1/R$
$F(8,7) = -\omega_z$
$F(8,14) = \sin(\alpha)$
$F(8,15) = \cos(\alpha)$
$F(8,17) = F_x \cdot \sin(\alpha)$
$F(8,18) = F_y \cdot \sin(\alpha)$
$F(8,19) = F_x \cdot \cos(\alpha)$
$F(8,20) = F_y \cdot \cos(\alpha)$
$F(8,23) = F_x \cdot F_y \cdot \sin(\alpha)$
$F(8,24) = F_x \cdot F_y \cdot \cos(\alpha)$
$F(8,26) = \omega_x \cdot \sin(\alpha)$
$F(8,27) = \omega_y \cdot \cos(\alpha)$
$F(8,29) = \Omega_z \cdot \sin(\alpha)$
$F(8,30) = -\omega_y \cdot \sin(\alpha)$
$F(8,31) = -\Omega_z \cdot \cos(\alpha)$
$F(8,32) = \omega_x \cdot \cos(\alpha)$

Table X f

$\underline{F}$ Matrix - Azimuth Error

Azimuth Error

$F(9,2) = \omega_N + \rho_Z x\tan(Lat)$

$F(9,3) = -\rho_Z/R$

$F(9,4) = \tan(Lat)/R$

$F(9,7) = \omega_N$

$F(9,8) = -\omega_E$

$F(9,16) = 1$

$F(9,21) = F_y$

$F(9,22) = F_Z + G_Z$

$F(9,25) = F_y \cdot (F_Z + G_Z)$

$F(9,28) = \Omega_Z$

$F(9,33) = \omega_y$

$F(9,34) = -\omega_x$

Table X g

$\underline{F}$ Matrix - Miscellaneous Error States

Vertical Acceleration Error

$F(11,3) = k_3$

$F(11,10) = -k_3 \cdot h$

$F(11,47) = -k_3$

Clock Phase Error

$F(12,13) = 1$

$F(12,52) = 1$

Clock Frequency Error

$F(13,51) = 1$

Baro-Altimeter Error Due to Variation in Altitude of a Constant Pressure Surface

$F(47,47) = -V/D_{ALT}$

East Deflection of Gravity

$F(48,48) = -V/D_{GE}$

North Deflection of Gravity

$F(49,49) = -V/D_{GN}$

Gravity Anomaly

$F(50,50) = -V/D_{GZ}$

Clock Random Frequency Error

$F(52,52) = -\beta_{tru}$

Table X h

$\underline{F}$ Matrix - Notes

The following states are modeled as random constants or random walks, and require no terms in the $\underline{F}$ matrix:

State Number	Description
10	Altimeter Bias
14-16	G-insensitivte gyro drifts
17-22	G-sensitive gyro drift coefficients
23-25	G^2-sensitive gyro drift coefficients
26-28	Gyro scale factor errors
29-74	Gyro input axis misalignments
35-37	Accelerometer Biases
38-40	Accelerometer Scale Factor errors
41-46	Accelerometer input axis misalignments
51	Clock aging bias

Values of Constant parameters

R = 20, 925, 639.76 feet

Ω = $7.292115147 \times 10^{-5}$ rad/sec

XK1 = .03

XK2 = .0003

XK3 = 1×10^{-6}

D_{GE} = 60,761.15 feet

D_{GN} = 60,761.15 feet

D_{GZ} = 364,566.9 feet

D_{ALT} = 1,519,028.75 feet

β_{tru} = 1/(1800 sec)

The terms listed here are for the truth model $\underline{F}$ matrix. The filter model $\underline{F}$ matrix consists of all listed terms for which both subscripts are less than or equal to 16.

Table XI

$\underline{Q}$ Matrix

The filter model $\underline{Q}_d$ matrices are derived in Chapter II.

For the truth model, a $\underline{Q}$ matrix is used to depict the driving noises. The terms listed here are actually terms of $\underline{G}\underline{Q}\underline{G}^T$, in order to show which state is driven by each of the noises.

$Q(12,12) = 100 \ (ft^2/sec)$
$Q(14,14) = 5.86 \times 10^{-20} \ ((rad/sec)^2/sec)$
$Q(15,15) = 5.86 \times 10^{-20} \ ((rad/sec)^2/sec)$
$Q(16,16) = 1.62 \times 10^{-19} \ ((rad/sec)^2/sec)$
$Q(35,35) = 2.88 \times 10^{-11} \ ((ft/sec^2)^2/sec)$
$Q(36,36) = 2.88 \times 10^{-11} \ ((ft/sec^2)^2/sec)$
$Q(37,37) = 2.88 \times 10^{-11} \ ((ft/sec^2)^2/sec)$
$Q(47,47) = 2.75 \times 10^{-2} \ (ft^2/sec)$
$Q(48,48) = 1.959 \times 10^{-8} \ ((ft/sec^2)^2/sec)$
$Q(49,49) = 8.375 \times 10^{-9} \ ((ft/sec^2)^2/sec)$
$Q(50,50) = 5.833 \times 10^{-9} \ ((ft/sec^2)^2/sec)$
$Q(52,52) = 2.77 \times 10^{-16} \ ((ft/sec)^2/sec)$

Note: As described in Reference 6, the noise terms for states 47-50 and 52 vary with velocity. However, the PROFGEN program keeps the path velocity constant throughout the flight profile described in Chapter IV. Therefore, the terms shown here were precomputed for the entire flight profile.

Table XII a

$\underline{H}$ Matrix - Notation

Each observation provides range and range rate measurements to each of four satellites. Measurement indices are:

k = satellite number (1-4)

i = range measurement to satellite k ($i=2 \cdot k-1$)

j = range rate measurement to satellite k ($j=2 \cdot k$)

Other notation:

$\bar{R}(k)$ = computed range to satellite k (feet)

$\dot{R}(k)$ = computed range rate to satellite k (ft/sec)

R' = range, receiver to earth center (feet)

$R_u(A_n,K)$ = Range unit vecotrs receiver to satellite k

n	Direction
1	East
2	North
3	Up

$\dot{R}_u(n,k)$ = Range rate unit vectors, reciever to satellite k

n	Direction
1	East
2	North
3	Up

L - Receiver Latitude

V_E, V_N, V_Z - Components of receiver velocity (ft/sec)

Table XII b

$\underline{H}$ Matrix - Equations

Range Measurements

$H(i,1) = R_u(1,k) \cdot R' \cdot \cos(L)$

$H(i,2) = R_u(2,k) \cdot R'$

$H(i,3) = R_u(3,k)$

$H(i,10) = 1$

Range Rate Measurements

$$H(j,1) = [\dot{R}_u(1,k) - R_u(1,k) \cdot (\bar{\dot{R}}/\bar{R}) \cdot R' + R_u(1,k) \cdot V_Z - R_u(3,k) \cdot V_E] \cdot \cos(L) + [R_u(2,k) \cdot V_E - R_u(1,k) \cdot V_N] \cdot \sin(L)$$

$$H(j,2) = [\dot{R}_u(2,k) - R_u(2,k)] \cdot (\bar{\dot{R}}/\bar{R}) \cdot R' + R_u(2,k) \cdot V_Z - R_u(3,k) \cdot V_N$$

$$H(j,3) = [\dot{R}_u(3,k) - R_u(3,k)] \cdot (\bar{\dot{R}}/\bar{R})$$

$H(j,4) = R_u(1,k)$

$H(j,5) = R_u(2,k)$

$H(j,6) = R_u(3,k)$

$H(j,11) = 1$

Table XII c

H Matrix - Evaluation

These H matrix terms were precomputed and used throughout the flight profile

Satellite Number(k)	1	2	3	4
H(i,1)	-1,284,390	-14,067,800	673,388	8,891,740
H(i,2)	-18,531,100	9,958,820	10,195,300	13,474,300
H(i,3)	.458117	.254804	.872353	.549816
H(i,10)	1	1	1	1
H(j,1)	77.8105	-1227.55	2179.51	-272.222
H(j,2)	1442.71	-2215.39	-1333.22	2028.61
H(j,3)	$.12888X10^{-3}$	$-.276022X10^{-4}$	$.322484X10^{-4}$	$-.916360X10^{-4}$
H(j,4)	-.076854	-.841773	.0402936	.532056
H(j,5)	-.885563	.475912	.487213	.643909
H(j,6)	.458117	.254804	.872353	.549816
H(j,11)	1	1	1	1

$i=2\cdot k-1$

$j=2\cdot k$

Table XIII

$\underline{R}$ Matrix

R(i,i) = Covariance of Range measurement if i is even.

= Covariance of Range Rate measurement if i is odd.

R(1,1) = 400 $(\text{feet})^2$

R(2,2) = .01 $(\text{ft/sec})^2$

R(3,3) = 400 $(\text{feet})^2$

R(4,4) = .01 $(\text{ft/sec})^2$

R(5,5) = 400 $(\text{feet})^2$

R(6,6) = .01 $(\text{ft/sec})^2$

R(7,7) = 400 $(\text{feet})^2$

R(8,8) = .01 $(\text{ft/sec})^2$

The $\underline{R}$ matrices for the truth model and filter model are identical.

Vita

David Arthur Arpin was born on 17 June 1950 in Norwich, Connecticut. He graduated from high school in Seattle, Washington in 1969 and attended the University of Washington, from which he received the degree of Bachelor of Science in Electrical Engineering in June 1973. Upon graduation, he received a commission in the USAF through the ROTC program. In August 1973, he was assigned to the Trajectory Division of the 544 Aerospace Reconnaissance Technical Wing of SAC at Offutt AFB, Nebraska. While at Offutt, he worked as a Trajector Engineer developing and maintaining Minuteman Missile Software. He entered the Air Force Institute of Technology School of Engineering in August 1976.

Permanent Address: 2040 N. W. 60th St.
Seattle, Washington 98107

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
14 AFIT/GE/EE/77-42

2. ~~GOVT ACCESSION NO.~~
9 Master's thesis

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
6 STATE NOISE COVARIANCE COMPUTATION IN THE KALMAN FILTER.

5. ~~TYPE OF REPORT~~ & PERIOD COVERED
MS Thesis

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 David ~~A.~~ Arthur Arpin
Capt ~~USAF~~

8. CONTRACT OR GRANT NUMBER(s)
12 150p.

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Institute of Technology (AFIT-EN)
Wright Patterson AFB Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
16 17 00
Project 7071-00-12

11. CONTROLLING OFFICE NAME AND ADDRESS
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433

12. REPORT DATE
11 Dec [redacted]77

13. ~~NUMBER OF PAGES~~
147

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17
Jerral F. Guess, Capt, USAF, Director of Information

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Kalman Filters
Inertial Navigation
Global Positioning System
Covariance Analysis
State Noise

Q sub d

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report investigates forms of the state noise covariance matrix in the Kalman Filter. This matrix, denoted Q_d, incorporates the effects of random errors driving the system dynamics into the filter computations. The Q_d matrix is derived by integration from the matrix of continuous time driving noise strengths, which normally includes only diagonal terms. This often leads to use of a diagonal Q_d matrix with constant terms. However, the derivation shows that Q_d should have off-diagonal and time varying terms.

→ next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

Q sub d

012 225

Block 20 continued

Q sub d

The study investigates the effects of including such terms in Q_d. Three alternate forms of Q_d are derived for a specific inertial navigation system. These, and a standard diagonal form, are tested using a covariance analysis. The results show little difference in performance for the different filters. This is attributed to two primary factors: highly accurate external measurements, and the use of integration sub-intervals for covariance propagation. These sub-intervals generate appropriate off-diagonal Q_d terms when a diagonal form of Q_d is used over each sub-interval. This suggests that an appropriate form of non-diagonal Q_d , which would not have to be added in at each sub-interval, could significantly reduce Kalman Filter computation requirements. Specific additional studies to test this possibility are suggested.